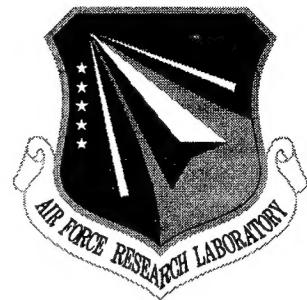


AFRL-IF-RS-TR-1999-127
Final Technical Report
June 1999



ARCHITECTURE AND NETWORK MANAGEMENT FOR NEXT-GENERATION INTERNET

Bellcore

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. F221

19990726 066

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-1999-127 has been reviewed and is approved for publication.

APPROVED: *Robert L. Kaminski*
ROBERT L. KAMINSKI
Project Engineer

FOR THE DIRECTOR:



WARREN H. DEBANY, Jr., Technical Advisor
Information Grid Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFGA, 525 Brooks Road, Rome, NY 13441-4505. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

ARCHITECTURE AND NETWORK MANAGEMENT FOR NEXT-GENERATION
INTERNET

Kenneth Young

Contractor: Bellcore

Contract Number: F30602-97-C-0263

Effective Date of Contract: 11 August 1997

Contract Expiration Date: 10 September 1998

Short Title of Work: Architecture and Network Management for Next-
Generation Internet

Period of Work Covered: Aug 97 - Sep 98

Principal Investigator: Kenneth C. Young, Jr.

Phone: (973) 829-4928

AFRL Project Engineer: Robert L. Kaminski

Phone: (315) 330-1865

Approved for public release; distribution unlimited.

This research was supported by the Defense Advanced Research
Projects Agency of the Department of Defense and was monitored
by Robert L. Kaminski, AFRL/IFGA, 525 Brooks Road, Rome, NY.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</p>			
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED	Final Aug 97 - Sep 98
4. TITLE AND SUBTITLE	ARCHITECTURE AND NETWORK MANAGEMENT FOR NEXT-GENERATION INTERNET		
6. AUTHOR(S)	Kenneth Young		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	Bellcore 445 South Street, MCC 1J206R Morristown, NJ 07960		
8. PERFORMING ORGANIZATION REPORT NUMBER	NGI A&BNM - 1		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	Defense Advanced Research Projects Agency 3701 North Fairfax Drive Arlington, VA 22203-1714		
10. SPONSORING/MONITORING AGENCY REPORT NUMBER	AFRL-IF-RS-TR-1999-127		
11. SUPPLEMENTARY NOTES			
AFRL Project Engineer: Robert Kaminski, IFGA, 315-330-1865			
12a. DISTRIBUTION AVAILABILITY STATEMENT		12b. DISTRIBUTION CODE	
Approved for public release; distribution unlimited.			
13. ABSTRACT (Maximum 200 words)			
<p>The goal of this project was to identify candidate network architectures and network management strategies for the Next-Generation Internet (NGI). For NGI network architectures, we first identified a set of Quality of Service (QOS) and reliability objectives to key NGI applications. We then qualitatively analyzed several NGI candidate backbone architectures based on several criteria: internetworking with end sites that use different technologies; internetworking between backbone subnetworks; routing switching; resource management; service restoration; and network configuration. We then completed a quantitative analysis of two specific WDM-based NGI backbone architectures (WDM alone and IP over WDM). We completed a detailed description of the function to be performed in the NGI backbone network and their relationship to one another. The comprehensive functional architecture that we have developed focuses on the case in which IP and WDM technologies are used in the inter-connected subnetworks. For NGI network management strategies, we developed integrated architectures for both the intra-domain and the inter-domain cases, with an emphasis on supporting QOS and survivability of key applications. We also developed a software architecture for an NGI intra-domain network system and specified the management interfaces between the subsystems using the CORBA Interface Definition Language (IDL).</p>			
14. SUBJECT TERMS		15. NUMBER OF PAGES	
Prototype System of Systems PSoS/Next Generation Internet (NGI), Quality of Service (QOS), Wavelength-Division Multiplexing (WDM)		76	
16. PRICE CODE			
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Abstract

The Next-Generation Internet initiative will “set the stage for the networks of tomorrow that are even more powerful and versatile than the current Internet.” Among the major challenges for the Next-Generation Internet is providing more consistent and reliable service quality that can be achieved today.

We report on a comprehensive study of potential Next-Generation Internet backbone network architectures and network management approaches, aimed at achieving the service quality levels required by future NGI applications. The network architecture studies resulted in the following output:

- A functional specification of a unicast IP/WDM backbone network architecture that supports the service quality levels required by NGI applications
- A functional specification of Autoconfiguration Points of Presence (AutoPops), which are interconnection points between the NGI site networks and the NGI backbone subnetworks, as well as between the NGI backbone subnetworks themselves
- Extension of the unicast IP/WDM functional specification to include multicast modes of communication.

The functional specification of the NGI backbone architecture includes a network management functional group. We developed a detailed description of this functional group, including the following:

- Integrated quality-of-service (QoS) management spanning the Application and IP layers that makes use of the capabilities of the underlying network architecture.
- Configuration management of IP/WDM networks.
- Inter-domain WDM network management.

This report, along with the associated deliverables referenced herein, summarizes our results.

Table of Contents

1. SUMMARY	2
2. INTRODUCTION	2
3. NETWORK ARCHITECTURE ALTERNATIVES FOR NGI	3
3.1. Service Quality Framework	3
3.2. NGI Applications and Network Requirements	3
3.3. Overview of Alternative NGI Backbone Network Architectures	5
3.4. NGI Backbone Network Functional Specifications	8
3.4.1. Information Transport Functional Group	9
3.4.2. Network Control Functional Group	11
3.4.3. Network Management Functional Group	13
3.5. Interconnection Point (AutoPop) Functional Specification	14
3.5.1. Network Element Information Transport Functional Group	15
3.5.2. Network Element Control Functional Group	16
3.5.3. Network Element Management Functional Group	17
3.6. Multicast NGI Network Functional Specification	18
3.6.1. Functional Architecture for IP-over-WDM Multicasting	19
3.6.2. Multicast Management Functional Group	20
3.6.3. Network Control Functional Group	21
3.6.4. Information Transport Functional Group	22
4. NETWORK MANAGEMENT APPROACHES FOR NGI.....	23
4.1. NGI Network Management Overview	23
4.1.1. NGI Network Environment	23
4.1.2. NGI Network Management Study Goals	24
4.2. Communications QoS Model.....	25
4.3. Proposed Application Layer QoS Model	26
4.3.1. Continuous Communication Class of Applications	26
4.3.2. Discrete Communication Class of Applications	28
4.4. IP Layer Communication/QoS Model	30
4.4.1. Review of Existing and Emerging Models	30
4.4.2. Proposed IP Layer Communication/QoS Model.....	32
4.5. QoS Management Architecture.....	36
4.5.1. Intradomain QoS Management Software Architecture	36
4.5.2. Subsystem Interfaces	39
4.6. Interdomain QoS Management Software Architecture	42
4.6.1. Proposed Solution Approach	42
4.6.2. Service Agreement between Neighboring Domains	42
4.6.3. Management Components and Interfaces	43
4.6.4. Management Components and Interfaces of the IPNMS	43
4.6.5. Merits of the proposed approach.....	49
4.6.6. Limitations of the Proposed Approach	49
4.7. A Comparison of QoS Mechanisms Used in ATM and Proposed NGI IPNMS.....	49
5. CONCLUSIONS	50
5.1. Network Architecture Results	50
5.2. Network Management Results	52
6. LIST OF ACRONYMS	55
7. REFERENCES	56

List of Figures

Figure 3-1 – WDM-only Backbone Subnetwork Architecture	6
Figure 3-2 – IP/WDM Backbone Subnetwork Architecture.....	6
Figure 3-3 – NGI Backbone Subnetwork Reference Architecture	8
Figure 3-4 – NGI Backbone Subnetwork Functional Architecture	9
Figure 3-5 – WDM Drop and Continue Technique	10
Figure 3-6 – Key Functional Components of AutoPop	15
Figure 3-7 – IP/WDM Multicast Functional Architecture	20
Figure 4-1 – Proposed Scheduling Policy.....	35
Figure 4-2 – Intradomain Network Management Architecture Components	37
Figure 4-3 – Management Interfaces	40
Figure 4-3 – End-to-end Setup of IP Pipes	43

List of Tables

Table 3-1 – Primary Characteristics of Target NGI Applications	4
Table 3-1 – Strengths and Weaknesses of WDM-only Backbone Subnetwork Architecture.....	7
Table 3-2 – Strengths and Weaknesses of IP/WDM Backbone Subnetwork Architecture.....	7
Table 3-3 - Strengths and Weaknesses of IP/ATM Backbone Subnetwork Architecture.....	7
Table 3-4 - Strengths and Weaknesses of IP/ATM/WDM Backbone Subnetwork Architecture	8
Table 4-1 – Application Layer QoS Parameters and Units.....	28
Table 4-2 – QoS Model for NGI Application Suite.....	29
Table 4-3 – Comparison of QoS Mechanisms Used in ATM with Proposed NGI IPNMS	50

1. Summary

We report on a comprehensive study of potential Next-Generation Internet backbone network architectures and network management approaches, aimed at achieving the service quality levels required by future NGI applications. The network architecture studies resulted in the following output:

- A functional specification of a unicast IP/WDM backbone network architecture that supports the service quality levels required by NGI applications
- A functional specification of Autoconfiguration Points of Presence (AutoPops), which are interconnection points between the NGI site networks and the NGI backbone subnetworks, as well as between the NGI backbone subnetworks themselves
- Extension of the unicast IP/WDM functional specification to include multicast modes of communication.

The functional specification of the NGI backbone architecture includes a network management functional group. We developed a detailed description of this functional group, including the following:

- Integrated quality-of-service (QoS) management spanning the Application and IP layers that makes use of the capabilities of the underlying network architecture.
- Configuration management of IP/WDM networks.
- Inter-domain WDM network management.

This report, along with the associated deliverables referenced herein, summarizes our results.

2. Introduction

The Next-Generation Internet (NGI) concept paper [1] describes the NGI vision as follows:

In the 21st Century, the Internet will provide a powerful and versatile environment for business, education, culture, and entertainment. Sight, sound, and even touch will be integrated through powerful computers, displays, and networks. People will use this environment to work, bank, study, shop, entertain, and visit with each other. Whether at the office, at home, or on travel, the environment will be the same. Security, reliability, and privacy, will be built in. The customer will be able to choose among different levels of service with varying prices. Benefits of this environment will include a more agile economy, a greater choice of places to live or work, easy access to life-long learning, and better opportunity to participate in the community, the Nation, and the world.

Clearly, realizing these ambitious goals will require significant research on many fronts. The goal of the work undertaken in this project was to identify potential NGI network architectures and network management approaches that can contribute to making this vision a reality. In particular, this work focuses on network architectures and network management strategies that will contribute to achieving the reliability and the service quality of the NGI referred to in this vision.

The report is organized as follows. In Section 3, we analyze potential NGI network architectures in light of the requirements of some of the more demanding NGI applications. Based on this analysis, we develop a functional specification for the NGI backbone network architecture, based on an IP/WDM approach. We also develop a functional architecture for the interconnection points to this backbone, which we refer to as AutoPops. We then extend the functional specification for the NGI backbone architecture to include multicast forms of communication.

In Section 4, we proceed to develop a network management approach intended to provide the quality of service (QoS) and reliability that NGI applications will demand and that the underlying network architecture can support. In developing the network management approach, we consider both the intradomain case, where all the NGI elements involved in an end-to-end application are under the control of a single administrative entity, and the interdomain case, where two or more administrative entities are

involved in an end-to-end application that must provide QoS and reliability. We summarize our major findings for NGI network architectures and network management in Section 5.

3. Network Architecture Alternatives for NGI

Given the plethora of emerging technologies for both high-speed user access and network backbone transport, one of the key challenges for NGI is identifying appropriate network architectures. The architectures, in turn, should be driven by the needs of the applications that NGI will support. Clearly, at a high level, two of the important characteristics of NGI applications that distinguish them from today's Internet applications are the need to support quality of service (QoS), such as bounds on delay and loss, and the need to provide better reliability/survivability than users currently experience. Therefore, in this project we have done both quantitative and qualitative studies of potential NGI architectures that assess their ability to support QoS and reliability/survivability.

Section 3.1 first discusses the assumed service quality framework for these studies and Section 3.2 discusses NGI applications and the network requirements derived therefrom. The results of the qualitative NGI network architecture study [2] are summarized Section 3.3. The results of the quantitative study can be found in [3]. Both studies form the basis for the functional requirements for the NGI backbone architecture and interconnection points discussed later in Sections 3.4, 3.5 and 3.6. Note that for the purposes of this report, we will use the term service quality to include both QoS and reliability/survivability.

3.1. Service Quality Framework

Broadly speaking, the spectrum of applications envisaged to be supported by the NGI can be categorized into the following two classes: (1) *discrete communication* class and (2) *continuous media communication* class. Examples of the former class include a request-response application model, supporting both unicast and multi-cast interactions and typically exist in a client-server scenario. The latter class of applications comprises continuous data exchange, of both unicast and multicast data, and typically exemplifies the producer-consumer scenario. We envisage that at the highest level, most of the data streams generated and used by applications will fall into one of these two classes.

Applications belonging to each of the above two classes may in turn require varying degrees of service guarantees (which we refer to as Quality of Service (QoS) assurances), from the underlying networks. Based on the QoS requirements, the applications in each of the above mentioned classes can again be categorized as: (a) those that do not require explicit QoS guarantees, and (b) those that require explicit QoS guarantees. Examples of category (a) typically consist of applications that receive best-effort service, e.g., web browsing with no explicit QoS assurances and unrestricted network access. Category (b) comprises application sets that are: (i) largely delay sensitive (e.g., voice and real-time video), (ii) largely loss sensitive (data, non-real time video images) and (iii) a mix of (i) and (ii) e.g., Global Command and Control (GCC) and distributed multimedia.

3.2. NGI Applications and Network Requirements

One of the prime objectives of the NGI initiative is developing novel network capabilities to enable a new wave of revolutionary applications [4]. Many of these emerging applications are motivated by DoD operations with requirements that include:

- Multimedia interactions with multi-point information exchange;
- High communication bandwidth with real-time constraints;
- Long session holding times possibly lasting several days (as in Synthetic Theater of War exercises); and
- Continuous, uninterrupted communications service (e.g., for mission critical operations).

We identified two key application types that are likely to impose particularly demanding requirements on the NGI backbone network: (1) *Distributed Interactive Simulations (DIS)*, which create a synthetic battlefield environment [5]; and (2) *Global Command and Control Systems (GCCS)*, which must be able to move a fighting force across the globe at any time while providing it with the information and direction it

needs to complete its mission [6]. Table 3-1 summarizes the primary characteristics of these two types of applications and their concomitant support requirements.

<i>DIS type of applications</i>	<i>GCCS type of applications</i>
<ul style="list-style-type: none"> Support for high-resolution multimedia computer-simulated war exercise sessions. Tight data coherency and synchronization between various CPUs due to strong data interdependencies. Low tolerance of data loss and data error. Information exchange in both unicast and multicast mode. Uninterrupted session holding time in terms of days. Support for fast reconfiguration of the participating units (adding, deleting, or moving units). Low tolerance for premature session release due to network failures. 	<ul style="list-style-type: none"> Support for multimedia video-conferencing sessions Medium tolerance of data loss and data error. Information exchange in both unicast and multicast modes. Uninterrupted session holding times in terms of hours. Support for fast reconfiguration of the troops. Support for high priority mission critical applications. Low tolerance for premature session release due to network failures in the case of high-priority, mission-critical applications.

Table 3-1 – Primary Characteristics of Target NGI Applications

In general, the network environment envisioned in the DARPA NGI program is that of a large-scale “network of networks,” nationwide in geographical scope and capable of supporting a wide range of demanding applications with equally wide-ranging Quality of Service (QoS) requirements. When combined with the characteristics in Table 3-1, this vision allows us to derive a set of key performance-related requirements. These requirements are elaborated below.

Support for sessions that require high bandwidth: Some DIS and GCCS applications need very high-resolution data transmission [5,6]. For instance, very detailed terrain information must be transmitted among participants in a war exercise in order to provide a realistic view of the battlefield. Whether for transmitting data associated with high-resolution images or a high quality video-conference, the implication is that support for high bandwidth sessions is required in the NGI backbone network.

Support for different delay tolerance: DIS applications are very dependent on the computational delays of associated remote units, and the distributed data must be synchronized in order to achieve the proper outcome. We therefore expect that these applications will require delay guarantees from the network in the low milliseconds range. However, for video-conferencing sessions such as those involved in GCCS applications, slightly longer delay guarantees (on the order of hundreds of milliseconds) may be permitted since the human ear is somewhat more tolerant than the distributed CPUs [7]. The delay requirements from DIS and GCCS applications can thus differ by almost orders of magnitude. This implies that the underlying backbone network may use its resources more efficiently by providing more than one level of delay guarantee, depending on the network technology used.

Support for different levels of loss tolerance: In the case of DIS applications, the need for transferring high-resolution images (such as battlefield terrain details) without much corruption translates into the need for low loss guarantees from the underlying network. By contrast, GCCS video-conferencing applications can, in general, tolerate more traffic loss. This again implies that the underlying backbone network may improve its resource utilization by providing different levels of loss guarantee, depending on the network technology used.

Support for unicast and multicast communications: Both DIS and GCCS types of applications are characterized by a combination of unicast and multicast sessions involving multimedia information. Thus, the underlying network should be capable of transporting multimedia information and supporting both point-to-point and multi-point communications.

Support for a wide range of session holding times (from hours to days): DIS applications have long holding times, typically spanning several days without interruption [5]. GCCS applications require

comparatively shorter holding times, and these different holding-time requirements have implications for network engineering. For example, while near-provisioning of resources can be done in order to fulfill the requested QoS for DIS applications, more dynamic approaches are needed for GCCS video-conferencing. An advantage in the former case is that relatively less stringent constraints are imposed on the routing strategies of the underlying network. This creates the potential for more optimal route searches, and therefore for more efficient use of network resources. In the case of GCCS video conferencing, however, efficient real-time dynamic routing strategies may be needed to find network paths that have enough resources to satisfy the application's requested QoS.

Support for flexible network reconfiguration and plug-and-play operation: Both GCCS and DIS applications involve movement of participating units as dictated by the battlefield environment (simulated or real), and these operations can drastically impact traffic load on a given portion of the network. The network will respond better to these changing demands if its resources can be reconfigured quickly. Support for plug-and-play operation of network equipment can contribute to the needed flexibility.

Support for service restoration times ranging from milliseconds to minutes: Service restoration mechanisms can be used to minimize the possibility of premature session release due to network failures. For applications that communicate via TCP streams/sockets, a restoration time in the 2-minute range is generally sufficient [8,9,10,11]. However, DIS applications and certain mission-critical types of GCCS applications may require shorter restoration times (in the order of seconds or milliseconds). This places different survivability requirements on the backbone network, which suggests the need for installing different restoration strategies.

3.3. Overview of Alternative NGI Backbone Network Architectures

As noted, the NGI backbone network is envisioned as a network of subnetworks, each of which may belong to a different administrative and/or routing domain. Among the central requirements that we outlined in an earlier deliverable [2] are support for:

- Interworking with NGI sites that may use different network technologies;
- Interworking between backbone subnetworks;
- Routing and switching;
- Resource management;
- Service restoration;
- Network reconfiguration.

Given these general functional requirements, four different subnetwork architectures for the backbone were evaluated, depending on the underlying transport technologies involved: (1) WDM-only, (2) IP/WDM, (3) ATM/WDM, and (4) IP/ATM/WDM. Thus, although the traffic generated by NGI sites is assumed to be all IP traffic, the NGI architecture could be composed of WDM, IP, and/or ATM network technologies.

In this section, we provide a brief overview of the four subnetwork architectures that were evaluated. In general, the alternatives can be classified as either single- or multi-hop lightwave network architectures. In a single-hop lightwave network architecture, information goes through no optical-to-electrical or electrical-to-optical conversion within the network; that is, only optical network equipment is used. By contrast, a multi-hop lightwave network architecture also includes network elements that operate in the electronic domain, and optical-to-electrical and/or electrical-to-optical signal conversion is therefore required [12,13].

WDM-only Backbone Subnetwork Architecture: This is a single-hop lightwave network architecture in which only WDM network equipment (WDM NEs) is used in the backbone subnetwork to transport information among the NGI sites (Figure 3-1). Information reaches the destination NGI site without going through any network equipment in the electronic domain, such as IP routers or ATM switches. The WDM NEs can be either WDM multiplexers (including WDM Terminal multiplexers/demultiplexers and WDM add/drop multiplexers) or WDM cross-connects.

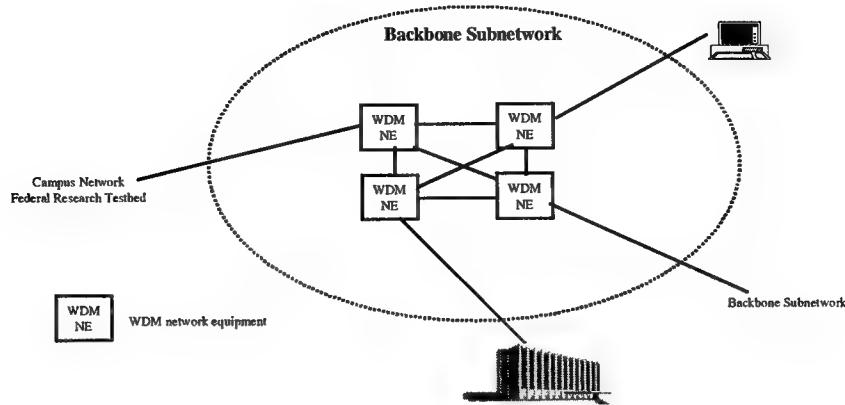


Figure 3-1 – WDM-only Backbone Subnetwork Architecture

IP/WDM Backbone Subnetwork Architecture: This is a multi-hop lightwave network architecture (Figure 3-2) that includes IP routers in addition to WDM NEs. NGI sites can be connected to high-speed IP routers, which in turn are connected via WDM NEs to facilitate communication among all NGI sites. Information may go through several IP routers before it reaches the destination NGI sites.

ATM/WDM Backbone Subnetwork Architecture: This multi-hop lightwave network architecture is similar to the IP/WDM architecture except that ATM switches are used instead of IP routers. NGI sites can be connected to these ATM switches, which in turn are connected via WDM NEs to facilitate communications among all NGI sites.

IP/ATM/WDM Backbone Subnetwork Architecture: This is again a multi-hop lightwave network architecture, but one in which NGI sites can be connected to either high-speed IP routers or ATM switches. In this case the high-speed IP routers are first connected to ATM switches. Those switches in turn are connected via WDM NEs to facilitate communications among all NGI sites.

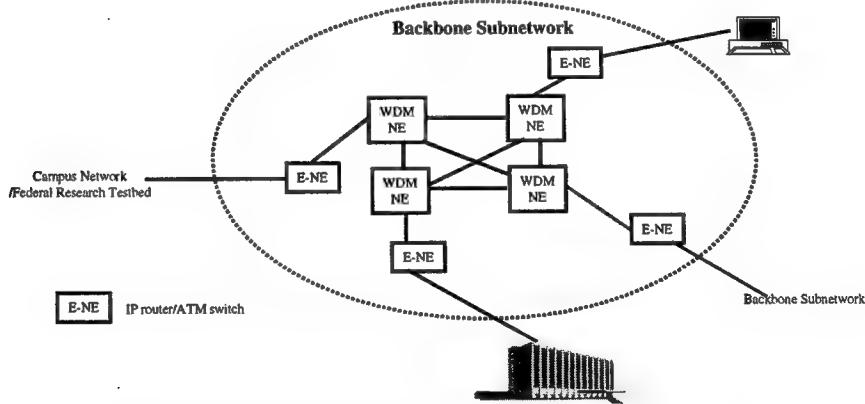


Figure 3-2 – IP/WDM Backbone Subnetwork Architecture

Tables 3-1 to 3-4 on the following pages summarize the strengths and weaknesses of each of these architectural alternatives in terms of its ability to support QoS, survivability, scalability, interoperability, and multicast communications. Comments on the conditions under which a given architecture is more desirable are also provided in these tables.

In general, the WDM-only backbone subnetwork architecture is of particular benefit when NGI sites are using different network technologies and transporting a mix of both IP and non-IP traffic. Further, when the WDM subnetwork can provide basic physical connectivity between the NGI sites but need not perform any processing of the transported information (e.g., no IP layer processing or the equivalent), the backbone

architecture is simplified overall. With this alternative, however, the NGI sites will have to maintain equipment for transmitting application layer traffic over WDM wavelengths, and many of the key technologies required for this are still early in development. Current technological limitations also constrain network size under this architecture.

On the other hand, the bandwidth requirements of most applications are significantly less than the gigabit per second capacity provided per wavelength in WDM. Therefore, another possibility is to use high-speed electronic switches or routers together with the WDM NEs in the backbone subnetworks. In this way, more intelligent processing can result in better resource sharing.

Strengths	Weaknesses
<ul style="list-style-type: none"> - Low delay and low loss during information transfer phase. - Transparent interoperability. - In principle, easier to scale to higher processing speed (when the required technology becomes available). <p>Comments: A desirable architecture if the number of NGI sites to be connected is not large and the bandwidth required between the sites is high.</p>	<ul style="list-style-type: none"> - Limited by the number of wavelengths supported per fiber, this architecture is difficult to scale up to connect a larger number of NGI sites. Even with future technology advances, wavelengths may still be limited resources when compared with IP addresses and ATM VPI/VCI values. - The technology is still in the early research and development stage.

Table 3-1 – Strengths and Weaknesses of WDM-only Backbone Subnetwork Architecture

Strengths	Weaknesses
<ul style="list-style-type: none"> - Improved survivability, scalability, and reconfigurability can be achieved via multi-layer approaches. - Easier to scale up to connect a larger number of NGI sites via interconnected routers. - More mature technology for multicast support. <p>Comments: A desirable architecture if the number of NGI sites to be connected is large, the bandwidth required between the sites is not in the gigabit per second range, and hard guaranteed QoS is not critical.</p>	<ul style="list-style-type: none"> - Scalability to higher processing speed is more of a challenge. - QoS support is a new concept in the connectionless IP networking paradigm. - Interoperability is not transparent. - Interworking standards between IP technology and other network technologies used on NGI sites needs to be developed. For instance, ATM signaling interworking support at an IP router.

Table 3-2 – Strengths and Weaknesses of IP/WDM Backbone Subnetwork Architecture

Strengths	Weaknesses
<ul style="list-style-type: none"> - Improved survivability, scalability, and reconfigurability can be achieved via multi-layer approaches. - Easier to scale up to connect a larger number of NGI sites via interconnected switches. - QoS support is embedded in the network architecture. For instance, ATM layer QoS signaling protocols exist now and QoS mechanisms are being incorporated in the ATM switches. - ATM multicast signaling standards exist and multicast mechanisms are being incorporated in the switches. <p>Comments: A desirable architecture if the number of NGI sites to be connected is large, the bandwidth required between the sites is not in the gigabit per second range, and hard guaranteed QoS is critical.</p>	<ul style="list-style-type: none"> - Even though ATM technology is designed for high speed networks using a hardware processing design paradigm, scaling up to a per port data rate of 10 gigabits per second or higher may present new challenges. - Complexity of network equipment and network operation may be a concern.

Table 3-3 - Strengths and Weaknesses of IP/ATM Backbone Subnetwork Architecture

<i>Strengths</i>	<i>Weaknesses</i>
<ul style="list-style-type: none"> Improved survivability, scalability, and reconfigurability can be achieved via multi-layer approaches. Easier to scale up to connect a larger number of NGI sites via interconnected IP routers or ATM switches. More mature technology for multicast support. <p>Comments: A desirable architecture if the number of NGI sites to be connected is large, the bandwidth required between the sites is not in the gigabit per second range, and hard guaranteed QoS is critical for some applications.</p>	May inherit both IP and ATM technology weaknesses.

Table 3-4 - Strengths and Weaknesses of IP/ATM/WDM Backbone Subnetwork Architecture

3.4. NGI Backbone Network Functional Specifications

Figure 3-3 depicts a high-level reference model of the NGI backbone. Note that this reference model consists of both single-hop and multi-hop lightwave subnetworks. The functional specifications provided in this report assume that IP/WDM technologies are used for the multi-hop lightwave subnetworks, but that the network technologies used for NGI sites connecting to the backbone can be very different. For instance, ATM-over-SONET can be used in one NGI site network and Gigabit Ethernet in another. The interconnection points between the NGI site networks and the NGI backbone subnetworks, as well as between the NGI backbone subnetworks themselves, are known as Auto-configuration Point of Presence, or *AutoPops*. The detailed functional architecture of an AutoPop will be provided in Section 3.5.

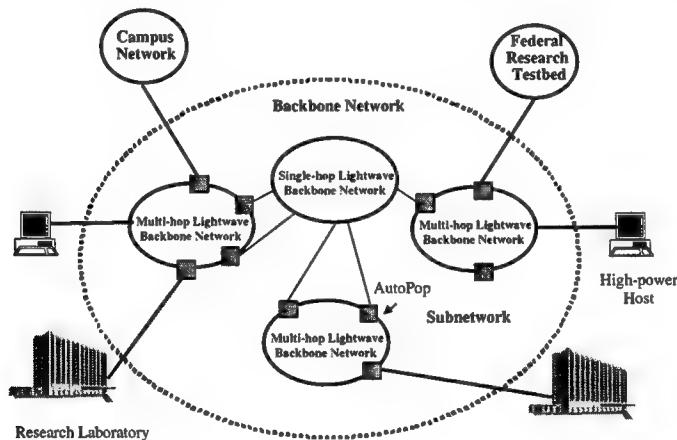


Figure 3-3 – NGI Backbone Subnetwork Reference Architecture

A functional architecture for the NGI backbone network is shown in Figure 3-4. This architecture is based on network capabilities for supporting QoS, survivability, scalability, and interoperability, as identified in our network architecture studies. Recall that an important requirement of the NGI backbone network is support for applications with high bandwidth requirements and real-time constraints.

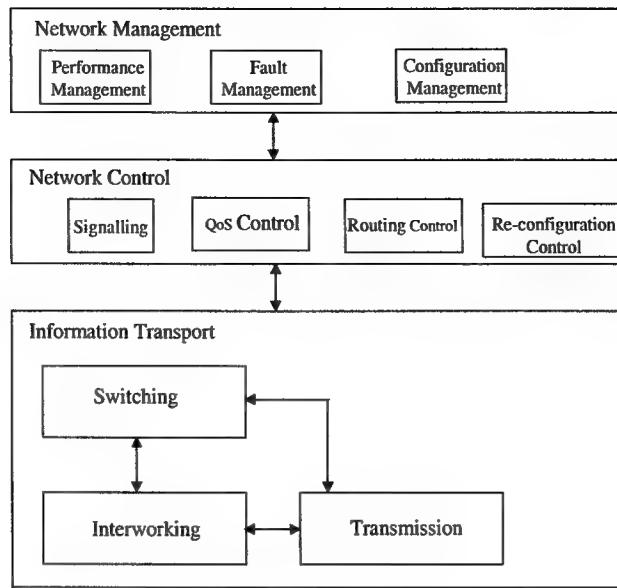


Figure 3-4 – NGI Backbone Subnetwork Functional Architecture

The functional architecture depicted in Figure 3-4 consists of three functional groups: (1) Information Transport, (2) Network Control, and (3) Network Management. The Information Transport group handles functions related to transporting information across the network. The key functional components in this group include transmission, interworking, and switching/routing. The Network Control group consists of functions related to real-time control of the network operations. Key network control functions include signaling, QoS control, routing control, and reconfiguration control. Finally, the Network Management group includes the key functions of network configuration management, network performance management, network fault management, and network security management. Network management functions interact closely with network control functions to ensure that information can be transported across the network with the QoS which the end applications require.

A description of each functional group identified in Figure 3-4 is presented below. An NGI network management architecture that defines the Network Management functional group is discussed in detail in Section 4 below. Therefore, in this section we will focus primarily on providing specifications for the Information Transport and Network Control functional groups.

3.4.1. Information Transport Functional Group

As noted, the Information Transport functional group consists of functions related to transporting information across the network. The key functional components in this group are transmission, interworking, and switching/routing, as described below.

Transmission

Ultra-high transmission capacity is a mandate if the NGI backbone network is to accommodate the bandwidth demands of emerging DoD applications [4], and this means that gigabit-level transmission capacity should be made available between NGI sites and some NGI site applications. To accomplish this, WDM technology should be utilized in the backbone to tap the virtually infinite capacity of the optics fiber. Observe that the NGI site networks may access the NGI backbone network via different transmission technologies on different media, such as high-capacity point-to-point wireless satellite links, optical fibers, and so on.

Physical interface specifications are an important element of the transmission function component specifications. These spell out the key functional characteristics for the underlying physical transport medium, including specifications for: (1) bit rate specifications; (2) formatting, framing and synchronization specifications; (3) fiber media characteristics; and (4) transmitter and receiver interface characteristics (which also include the wavelength (signal) adding/dropping interface). Note that the bit streams associated with IP datagrams in the electrical domain must be converted to optical signals before they are handed off over the physical (optical fiber) medium.

Recently, a group at USC has proposed the use of Smart WDM IP Flow Technology (SWIFT) for IP networks that support native optical transmission (i.e., the IP-over-WDM networks discussed in this report) [14]. In essence, SWIFT is designed to use intermediate, smart optical processing to augment the relatively low-bandwidth and high-latency IP forwarding software on the one hand, and the typically high-bandwidth, lower-latency but less flexible all-optical switching on the other. The development goal is the “best of both worlds,” providing more flexibility than an all-optical path yet better throughput than an all-electrical path.

To accommodate SWIFT, the transmission function component should also include dynamic dispersion compensation and optical contention detection to support dynamic contention resolution, among other things. A survivability benefit is that redundant transmission capacities can be provisioned to minimize the impact of network failure on network services, via certain automatic protection switching systems [15]. Further, the inherent broadcast nature of optical devices like the optical coupler and optical power splitter can be used to provide network monitoring and control support at the WDM layer in the Transmission function component. For instance, a technique called *Drop and Continue* (Figure 3-5) drops a portion of an incoming optical signal and passes the remainder on to an outgoing optical interface. The dropped optical portion of the signal and the continuing portion can be restored to the original signal’s magnitude by optical amplifiers. Network traffic can thus be passively tapped at the WDM layer in the optical power splitter, and then analyzed to determine the current network status.

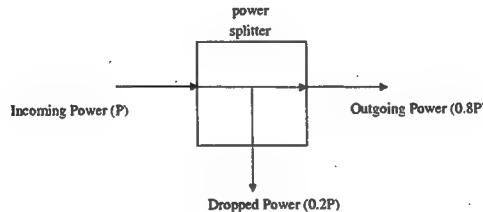


Figure 3-5 – WDM Drop and Continue Technique

Interworking

If NGI sites employ different network technologies, the NGI backbone network must support interworking functions in order to provide ubiquitous communications. Such interworking functions include performing protocol conversions in such a way that, within a common layer service, the protocol information of one protocol is extracted and mapped onto the protocol information of another.

The common layer service is defined by the functions common to the two protocols involved. Recall that all NGI applications are assumed to use IP services provided by the network layer in the OSI protocol reference model. However, when different NGI sites use different network technologies, the common layer service is provided at the IP layer (i.e., protocol conversion is done below the IP layer).

This function component will also include the interworking of signaling between the NGI site networks that use ATM technology and the IP-based NGI backbone networks. Interworking is used to allow ATM-based NGI sites to communicate with each other via the IP-over-WDM-based backbone network.¹

¹ If native ATM services are supported on some NGI sites, additional interworking functions are needed to allow end systems using ATM services to communicate with end systems using IP services.

The interworking function is one of the key features of the AutoPops (the interconnection points between the NGI site networks and the NGI backbone subnetworks). The functional specifications for an AutoPop will be detailed in Section 3.5.

Switching

Switching in the Electronic Domain

In the electronic domain, switching is the process of forwarding packet data at an intermediate system in the network, and each packet typically traverses at least one such intermediate system before reaching its intended destination. If a pre-selected route is specified with the packet, the intermediate system forwards the packet accordingly. Otherwise, it examines the packet's intended destination and selects the forwarding route locally.

This function interacts with components from different functional groups. For example, note that the forwarding process requires scheduling and buffer management mechanisms, and the way in which these mechanisms operate is determined by the QoS control function component of the Network Control functional group (see Figure 3-5). Similarly, the switching function that resides in an AutoPop also involves traffic metering. It controls traffic entering the network by marking or discarding the data units (i.e., the packets) based on instructions it receives from the QoS control function component [16]. Finally, the routes from which the switching function component can select are determined by the routing control function component in the Network Control functional group. Note that the switching and routing functional components can interact dynamically to select an optimal route. Specifically, the switching component can inform the QoS component based on the statistics collected by the buffer management mechanism. The QoS component may in turn trigger new route calculations by the routing control function component, and/or reconfiguration of router connections or connection capacities by the control function component.

Switching in the Optical Domain

In the optical domain, switching is the process of forwarding optical signals via wavelength switching in an optical intermediate system. To facilitate switching in the optical case, the switching function component may convert a received optical signal to a different wavelength before forwarding the signal toward its intended destination.

In addition to dynamic wavelength translation, all-optical header modification can also be performed by the switching function component in order to support flow switching (via modified flow switching techniques). The wavelength and input/output port mappings used by this switching function component are determined by the configuration management and reconfiguration control function components. Note that the reconfiguration control function component interacts with QoS control to reassign this mapping dynamically. The reconfiguration control function component may also trigger the routing control function component to recalculate routes.

3.4.2. Network Control Functional Group

As noted, the Network Control functional group consists of functions related to the real-time control of network operations. Key network control functions include signaling, QoS control, routing control, and reconfiguration control. A description of each of these functions follows.

Signaling

The signaling function component includes the processing of QoS signaling, routing protocols (both intra-domain and inter-domain), and network reconfiguration signaling. This function component supports communications among different network elements to convey the information required by QoS control, routing control, and reconfiguration control function components.

QoS signaling

Two major approaches (Integrated Services/RSVP and Differentiated Services) for supporting QoS in IP-based networks are being pursued in the IETF [17,18,19,20,21]. Although these two approaches differ in the nature of the QoS guarantees they offer, both can be used in an IP-over-WDM backbone network architecture to adequately support the required QoS.

One of the key components supporting QoS in the NGI backbone network is QoS signaling, either between the routers or between the network and the end applications. QoS signaling is the mechanism used to negotiate the QoS a data unit requires from the network, and to convey what QoS requirements in turn that the network places on the data. In an IP-over-WDM backbone architecture, there must be a mechanism for signaling the QoS required by a packet in order to support QoS at the IP layer using the Differentiated Service model. One approach is to use the IP packet header field for indicating the requested service class to the NGI backbone network via an AutoPop, and to have the packet classification be determined by the NGI backbone network [16,19]. In this approach, packet classification would be performed at the AutoPop as part of the traffic metering function, and be based on the provisioned service class profile.

Further, in order to provide a requested QoS, the IP-over-WDM based NGI backbone network must also be able to deny the establishment of a new session. This implies the need for supporting QoS signaling associated with new session establishment. To accomplish this, the packet's QoS requirement can be conveyed to the QoS control function component, which can then (a) determine whether or not a new session can be established, and (b) control the corresponding operations of the switching function component in the Information Transport functional group. QoS signaling capabilities can also be used to facilitate QoS negotiation between the network and the end-application via the session management function entity.

QoS routing is another important component that holds great promise for providing QoS in NGI backbone network, and has just started to gain interest in the research community. To facilitate dynamic QoS routing, the capability for QoS signaling also needs to be incorporated among the routers. For instance, should the routers' QoS support capabilities change with time, QoS signaling may be used for negotiating the QoS commitments provided to different service classes.

Routing Protocols

A routing protocol provides a mechanism for distributing route update messages. The routing control function component uses information conveyed by the routing protocol to determine which route will be used by the switching function component (in the Information Transport functional group). Many dynamic routing protocols are in use today in IP-based networks, including the OSPF (used on an intra-domain basis) and the BGP (on an inter-domain basis) [22,23]. The implications of using these same protocols in a gigabit NGI backbone network environment require further investigation.

Reconfiguration Signaling

The underlying WDM layer can be exploited for the QoS architecture by designating high-performance optical paths between the routers. QoS flexibility can then be enhanced by allowing for the dynamic reconfiguration of these optical channels or for establishing them on demand for certain supported applications. Reconfiguration signaling provides a mechanism for dynamically establishing such optical channels via the reassignment of wavelengths in the NGI backbone network. The information conveyed in the reconfiguration signaling messages is used by the reconfiguration control function component to carry out the network reconfiguration operation.

QoS Control

For an IP-over-WDM-based NGI backbone network to provide adequate QoS, additional mechanisms need to be implemented at the IP routers/switches. These include mechanisms for traffic profile specification, traffic metering and enforcement, intelligent scheduling, buffer management, and resource allocation. These mechanisms are implemented in the Information Transport group (as part of the switching function component); their parameters can be determined by the QoS control function component in interaction with the Network Management functional group. For example, the QoS control component interacts with the Network Management functional group to determine the service class profile, the corresponding packet

classification, and the associated actions to be taken by the traffic metering mechanism (in the Information Transport functional group).

Routing Control

The routing control function component deals with the routing algorithm that is used to calculate optimal routes for the data. A “best route” calculation will depend on features of the routing protocol that determine which metrics will be available to the routing control function. It will also depend on the relative weights assigned to those metrics, which in turn are influenced by the routing policy imposed on the NGI backbone network.

The routing control function component should implement a routing algorithm that will perform correctly under unusual or unforeseen circumstances, such as unexpected traffic surge or network failure. Further, the routing algorithm should be able to support multiple routes to the same destination, in order to provide better network throughput and survivability.

Both intra-domain and inter-domain routing needs to be supported by the NGI backbone network. Note that inter- and intra-domain routing are of very different natures. Intra-domain routing attempts to maximize the efficiency of network utilization, using detailed network information to find optimal routes. By contrast, inter-domain routing uses aggregated domain information to find routes that in many cases are dictated by administrative policy or algorithm complexity.

Reconfiguration Control

As described earlier, it may be desirable to establish optical channels on a dynamic, on-demand basis for certain supported applications, or for paths between certain routers within the NGI backbone. The reconfiguration control function component handles this as part of its responsibilities for reassigning router connectivities and inter-router bandwidth. Its action can be triggered by network management functions residing in the Network Management functional group.

For instance, reconfiguration control can be initiated to reassign the capacity between certain connected routers within the NGI backbone network based on performance statistics collected by the network performance management function component (which is part of the Network Management functional group). In addition, the reconfiguration control component may be triggered after network facility failures in order to minimize service interruption. Note that provisioning-based APS capabilities can also be used at the WDM layer to improve NGI backbone network survivability.

In an IP/WDM network, the reconfiguration strategy may reside entirely within the WDM layer, entirely within the IP layer, or in both layers. In Appendix A of [24], we discuss some of the possible reconfiguration strategies that are applicable to an IP/WDM NGI backbone network in more detail. Appendix A also includes covers the functional partitioning of a multi-layer reconfiguration strategy, and the corresponding service conditions that may be best suited to this functional partitioning. Finally, some performance limitations of the reconfiguration strategy options are briefly summarized.

3.4.3. Network Management Functional Group

The Network Management functional group consists of management-related functions that interact closely with network control functions to ensure that information can be transported with the QoS required by the end applications. Network fault management, network performance management, and network configuration management functions reside within this group (see Figure 3-4).

Fault Management

One key responsibility of the fault management function component is the coordination/mediation that is needed when a network fault is detected. Usually, the failure monitoring and detection processes will inform the fault management function component whenever a network facility failure occurs. Once notified, the fault management function component will take appropriate actions (i.e., log the failure event for statistical purposes and possible maintenance), and then trigger any applicable reconfiguration control functions (as in the case of self-healing or survivable/robust networks). The particular reconfiguration mechanism may reside either entirely within the IP layer, the WDM layer, or in both layers (as elaborated

in Appendix A of [24]). But the fault management function component will be responsible for coordinating the triggering of appropriate reconfiguration mechanisms.

Performance Management

The performance management function component monitors the network links for both usage and condition (i.e., whether they are up or down). This component may trigger the reconfiguration control function to reconfigure or change the capacity of the underlying WDM network (by adding wavelengths) if, for example, it detects congestion based on the performance statistics collected. Another functionality that may be incorporated within the performance management component is support for the QoS control and routing control function components, in order to facilitate session management (e.g., session establishment).

Configuration Management

The configuration management function component is responsible for managing functions related to network configuration, such as network topology, router connectivity, inter-router link capacity, and reconfiguration control strategies. It may receive information from the fault management and/or performance management function components in order to update the network configuration data that is maintained. Further, network topology, router connectivities, and inter-router link capacities may be changed as a result of these network state updates. The configuration management function component differs from the reconfiguration control function component in that the latter operates in real-time whereas configuration management, in general, operates in non-real-time mode.

3.5. *Interconnection Point (AutoPop) Functional Specification*

Recall that the AutoPop is an inter-connection point between the NGI backbone subnetworks themselves, and between those subnetworks and the attached NGI site networks. One of the key functionalities of an AutoPop is to support interworking between NGI backbone subnetworks and the attached NGI site networks. In this section, we consider the functional specifications for AutoPops in the NGI backbone network architecture.

Figure 3-6 depicts the key functional components of an AutoPop, including auto-configuration, fault detection, performance monitoring, QoS control, signaling, routing control, mobility management, reconfiguration control, switching, interworking, and transmission. These functional components are classified into three functional groups: namely, (1) Network Element Management; (2) Network Element Control; and (3) Network Element Information Transport. The characteristics of each group are discussed below.

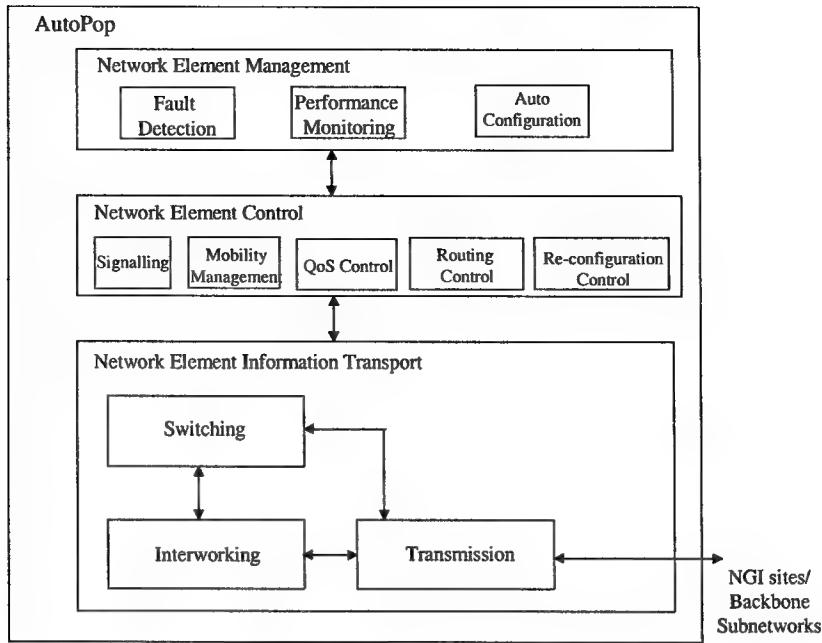


Figure 3-6 – Key Functional Components of AutoPop

3.5.1. Network Element Information Transport Functional Group

The Network Element Information Transport group includes three function components related to receiving and forwarding data units at the AutoPop: (1) transmission; (2) interworking; and (3) switching. A brief description of each follows.

Transmission

The transmission function component of the AutoPop performs basically the same functions as those described in Section 3.4. It is responsible for receiving signals from the transmission medium (wireless links, optical fibers, etc.) and converting those signals into bit streams between the NGI sites and the NGI backbone network. This function component is also responsible for extracting the data units from the bit streams that it receives.

Interworking

Interworking is required because different network technologies may be employed among the NGI sites attached to the NGI backbone network. The interworking functions described in Section 3.4 are primarily performed in the interworking function component of an AutoPop. AutoPops that connect NGI site networks to backbone subnetworks need to support interworking with the network technology that a particular site uses. For instance, if ATM technology is used in an NGI site network, this component of the AutoPop is responsible for signaling interworking between that site network and the NGI backbone network. In particular, interworking functions will be required in this case to map ATM signaling messages to the corresponding information needed for transporting IP packets.

Switching

The switching function component is responsible for receiving a packet, determining where to forward it, scheduling the queued packet for service, and performing access control and buffer management. Its functions are performed in accordance with instructions from the QoS control function component. The switching component in an AutoPop includes all the functions described in Section 3.4. Indeed, it is primarily the AutoPop's switching function component that performs traffic metering and access control at the session layer as described in Section 3.4.

3.5.2. Network Element Control Functional Group

The Network Element Control functional group consists of functions related to the real-time control of AutoPop operations, including QoS control, signaling, routing control, reconfiguration control, and mobility management. A description of each of these functions follows.

QoS Control

For a backbone network based on IP-over-WDM technologies, network access control functions can be used to limit traffic admitted into the network at AutoPops in order to provide a given level of QoS to the network users. Such network access control functions can be implemented at the packet level and/or at the session level. For instance, the traffic metering being proposed in IETF is one example of a packet-level network access control mechanism. By contrast, the blocking of new sessions based on current network states is an example of network access control functions at the session level.

The QoS control function component will coordinate with the switching function component within the AutoPop to provide requested QoS to NGI network applications. For instance, the QoS control component will determine the scheduling priority for a given supported service class, and instruct the switching/routing functional component to serve the arriving packet accordingly. To perform some of these functions, the QoS control component may need to communicate with other peer entities via the signaling function component. Further, if the connected networks (either two backbone subnetworks or an NGI site network and a backbone subnetwork) are using different network technologies or providing different network performance, procedures will be needed to map the network performance requirements.

To facilitate network performance management, the QoS control function component may include active network capabilities that enable customized collection of QoS statistics for the wireless access link. These QoS statistics could be used, for example, to trigger reconfiguration of the wireless access link in order to reassign the radio frequency, time slot, or transmission rate. Further, via mutual service agreement the QoS control function component may perform customized QoS monitoring functions for the wireless access network. As a provided customer service, the NGI backbone network might use such monitoring statistics to make reconfiguration recommendations to the wireless access network in order to minimize the impact of projected network problems. In general, the issues and implications associated with inter-administrative domain management need further investigation.

Signaling

Signaling is required for information exchange among entities that must cooperate in the performance of a specific function. For example, QoS signaling is needed between the network elements and the AutoPops to facilitate QoS routing. It is also needed to indicate or negotiate the requested QoS with the backbone network. Finally, signaling functionality is required to facilitate information exchange in support of auto-configuration, routing control, and fault management functions.

Routing Control

The routing control function component of the AutoPop deals with the routing algorithm used to determine the best route for forwarding data. Both intra- and inter-domain routing need to be supported at the AutoPop. The QoS control function component and the fault management function component can trigger the routing control component to recalculate routes for use by the switching function component.

Reconfiguration Control

The reconfiguration control function component of the AutoPop provides self-healing capabilities among AutoPops that connect the backbone subnetworks, in order to minimize network downtime due to network failures. Depending upon the nature of the detected fault, reconfiguration of the subnetwork connectivity via optical channel reassignment may be initiated. Note that reconfiguration at the IP layer may be performed to reroute traffic without initiating network reconfiguration at the WDM layer (i.e., without changing router connectivities and/or inter-router capacities via reassignment of wavelengths).

This self-healing capability may also facilitate the addition or removal of AutoPops without service interruption. In general, being able to dynamically reconfigure backbone subnetwork connectivity via optical channel reassignment and AutoPop addition/removal adds a new dimension to network survivability

support. Optical channel reassignment may also be used to increase the bandwidth between two connected backbone subnetworks.

Mobility Management

We envision that the AutoPop should include mobility management functions to support on-the-move broadband wireless services in the NGI backbone network. For instance, an NGI site could be a local area network on an airplane, which is connected to other NGI sites via the AutoPops on the NGI backbone. This NGI site network could use low-orbit satellites as relays to move from one AutoPop to another without losing communication. To enable such a scenario, the mobility management function uses two databases to keep track of the locations of an on-the-move NGI site network: namely, a Home Location Register (HLR) and a Visitor Location Register (VLR). Two of the key capabilities required in the AutoPop to support the mobility management function are location registration/update and hand-off.

3.5.3. Network Element Management Functional Group

The Network Element Management functional group consists of network management related functions that fall within the scope of an AutoPop. These functions interact closely with the network control functions at the AutoPop to ensure proper processing of the information it receives. Note that the backbone network's ability to deliver the QoS required by end applications depends closely on the access control performed at the AutoPop. A brief description of the AutoPop's auto-configuration, performance monitoring, and fault detection function components follows.

Autoconfiguration

Being able to automatically configure the parameters required for connecting the NGI site networks to the backbone subnetwork will ease the access setup process for the NGI backbone. The parameters that need to be configured include: (1) the network address of the NGI site network; (2) the network protocol and the corresponding options used across the access interface; and (3) the link layer protocol and the corresponding options used across the access interface. Note that an automatic address configuration procedure has yet to be developed.

The access interface to NGI backbone network from NGI site networks may be a wireless link. In this case, the auto-configuration function component is responsible for configuring the radio link to be used. The proper network configuration in this case is closely related to the wireless access technology used. Further, the topologies of most wireless networks are dependent upon the surrounding environmental conditions. Local information such as building height, building density, foliage density, and amount of local rainfall can all impact the topology of a wireless access network. The auto-configuration function component can be used, via specific service agreement, to help a mobile NGI site network establish its network topology (connectivity and capacity) and to facilitate the NGI site networks' plug-and-play operation. For wireless access networks, then, auto-configuration capability involves:

- Maintaining in a database system that local information which is relevant for configuring the wireless network. Which information should be considered relevant is closely related to the wireless technology used. For instance, if some of the broadband wireless technologies are used where line-of-sight operation is critical, terrain information such as the distribution of high buildings and other tall objects may be important.
- Enabling automatic discovery of the specific technology used in a newly attached wireless access network.
- Generating network topology (including connectivity, location, and link capacity) for the wireless access network based on the wireless access technology used and the local information maintained in the database system. This topological information will facilitate plug-and-play operation for the wireless access network.

Because an AutoPop can also be an inter-connection point between backbone subnetworks under different administrative domains, the auto-configuration component will coordinate with the network management systems (specifically, network configuration management) of the networks involved to determine the policies related to routing, QoS support, and fault management. Again, this implies that signaling

capabilities are needed to facilitate communications with the network management systems and the peer auto-configuration functional components.

Fault Detection

Fault detection is a key component of the network fault management function. This component detects abnormal conditions based on certain monitored data. It generates notification messages according to a given set of criteria, and determines whether or not to trigger reconfiguration mechanisms. Depending upon the nature of the detected fault, reconfiguration of the subnetwork connectivity via optical channel reassignment may be initiated.

Performance Monitoring

The performance monitoring function component monitors the states and usage of the various resources at the AutoPop, and is also responsible for collecting and processing performance data. Based on the performance statistics it collects, this component may inform the QoS control, routing control, and reconfiguration-control function components of significant performance changes.

3.6. Multicast NGI Network Functional Specification

In general, three key functionalities are required for a successful implementation of multicasting [25,26,27]: (1) multicast protocols for establishing and managing multicast groups; (2) network elements (*i.e.*, switches *or* routers) that can coordinate multicast protocols in order to distribute multicast messages and manage groups; and (3) an application protocol and APIs (Application Programming Interfaces) for initiating and facilitating multicast sessions. These functional requirements assume that end-to-end transport protocols are available for the orderly transmission and reception of message streams for target applications, and that there is a network infrastructure in place to provide the basic means for network communication.

In addition, however, the internetworking environment that arises when different network technologies and administrative strategies are merged into a heterogeneous network domain introduces challenging issues of interoperability among different multicast routing protocols. The scalability problems that can result are among the primary reasons why multicast protocols are still in a state of continuing evolution. Overall then, defining an architectural framework for supporting multicast in the NGI network model – that is, in a network of subnetworks that can use different network technologies – dictates that interoperability be carefully considered throughout every phase of system design.

Additional challenges to the seamless integration of WDM with existing network infrastructures stem from physical limitations inherent in current WDM technology. These limitations pertain to reconfiguration times that are slow in relation to the high bandwidth that WDM offers, and to the available number of wavelengths per optical fiber. Although the shortcomings – mostly governed by the laws of physics - will improve to some extent over the coming years, they are not likely to be completely resolved. The evolution of WDM technology in the next generation will therefore be centered on coping with them.

One of the most important aspects of a multicast communication paradigm is the capability for efficiently managing multicast groups. Most of today's multicast protocols are designed to enable dynamic group management, in which hosts can freely come and go during a multicast session. Dynamic group management requires network elements to update their states promptly to reflect the ever-changing set of hosts in a multicast group. This calls for localized state updates in the routers or switches affected by the changes to facilitate rapid state adjustments. The concept of IP multicasting is based on the IP networking paradigm in general, where network elements (*i.e.*, routers) are often designed to react quickly to local state changes without necessarily consulting global information. Therefore, the IP environment can support dynamic group management without much difficulty. By contrast, WDM has a rather complicated reconfiguration (switching) mechanism that depends on centralized, global network state information. This results in generally slow reconfiguration times, and thus makes WDM poorly suited for dynamic group management.

Multicast groups can also become very large. This is especially true in a distributed simulation environment (which includes academic work on multiparty games and DoD work on global battle simulations²), in which literally tens of thousands of hosts may be involved [24]. A WDM implementation for supporting multicast applications like these is destined to fall short of the number of wavelengths it needs to provide every host with a communication channel.

In summary, the current state of WDM technology falls short of the ideal when it comes to support for the emerging breed of applications that rely on multicast communication. Resource utilization and scalability issues are among the major challenges that multicast presents to today's WDM networks – especially multicast with dynamic group membership. At the same time, however, the challenges inherent in the design of WDM-layer multicast protocols are likely to stimulate further technological advancements.

In this section, we focus on the internetworking aspects of the WDM subnetwork technology (based on IP-over-WDM) that are relevant to providing support for multicast communication in the NGI network. We first present a functional architecture for IP-over-WDM multicast in the NGI system. We then propose a suite of multicast routing architectures suitable for the WDM subnetwork, each of which deserves further investigation for developing a high-performance, multicast-capable NGI network system. Each routing architecture takes into consideration the limitations (*e.g.*, reconfiguration delay and number of available wavelength per fiber) and advantages (*e.g.*, high communication bandwidth and built-in multicast-capable switch functions such as *drop-and-continue*) of both current WDM technologies and future developments. In addition, for each proposed routing architecture we delineate the sets of functions that need to be augmented or modified from their definitions given in Section 3.4 for the unicast communication paradigm. Finally, we conclude this section by discussing the strengths and limitations of a multicast-layer interworking architecture, focusing on potential implications for the unicast functional model.

3.6.1. Functional Architecture for IP-over-WDM Multicasting

Figure 3-7 presents the functional architecture proposed for IP-over-WDM multicasting in the NGI network. The functional groups for this multicast architecture are Information Transport, Network Control, Multicast Management, and Network Management. The Information Transport, Network Control, and Network Management functional groups have been described in detail in Section 3.4 for unicast communication. In this section, we will outline any functional changes to these groups that are required for supporting multicast communication. However, we will focus in most detail on the new Multicast Management group, in which an entirely new set of functions is introduced for the implementation of a multicast capable NGI network.

² The DoD goals are to use IP multicast for simulations that involve over 10,000 simultaneous groups for upwards of 100,000 simulation processes (virtual entities) in a global-sized WAN by the year 2000.

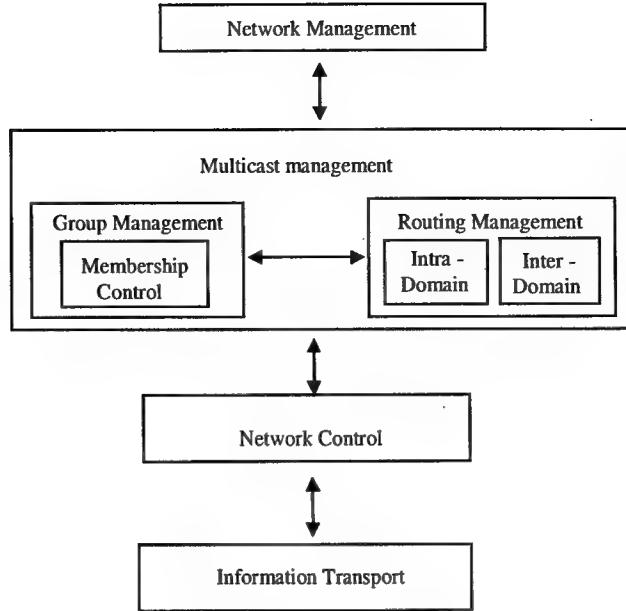


Figure 3-7 – IP/WDM Multicast Functional Architecture

3.6.2. Multicast Management Functional Group

The main vehicle for NGI multicasting is provided by a group of multicast management functions that interact with the Network Management, Network Control, and Information Transport functional groups (either directly or indirectly) in the NGI function hierarchy. The key functional components in this group consist of multicast group management and routing management, each of which interacts with the others and consists of the sub-components identified in Figure 3-7.

Group Management

One of the main functional components of multicasting is group management, which is the mechanism that determines how hosts join and leave a multicast group. Multicast groups can be either static or dynamic. Once a static group has been set up, its membership remains unmodified until the group is discarded; *i.e.*, a static group stays fixed in a multicast session. By contrast, the membership of a dynamic group can grow or shrink during a session.

Though it offers the flexibility for hosts to join and leave a group freely, dynamic group membership creates the need for tracking active members, and thus introduces more complex group management requirements. For example, routers require added functionality to update the group membership of the hosts they serve in a subnetwork as the hosts leave or join the group. When a centralized list of group members is not available in the network, dynamic group membership forces the routers to periodically poll their subnetworks in order to determine which ones contain group members. In addition, the AutoPops at domain borders in NGI network need to have functions for relaying membership information to adjacent subnetworks along the multicast tree each time membership is updated. Finally, dynamic group membership can negatively affect network bandwidth -- the very resource multicasting is trying to conserve -- due to the delays required for reconfiguring the multicast tree each time a host leaves the group.

Despite these drawbacks, group membership is nonetheless managed dynamically under current Internet protocols. In the current IP multicast environment, joining and leaving a multicast host group is done through a signaling protocol called the Internet Group Management Protocol (IGMP) [28], which periodically issues a query to the host group to update host group membership lists.

Unfortunately, however, the WDM layer cannot currently support dynamic group management due to its slow reconfiguration response times. Though some of the strategic DoD applications may not require dynamic membership management, the goal of smooth integration among NGI subnetworks with different

network applications and management strategies makes it desirable for the WDM network to have this capability. Much effort is being put into developing fast re-configurable WDM switches based on opto-electronic processing. For now, however, this work is still too premature to enable the implementation of fully dynamic membership management in the WDM domain.

Routing Management

This functional component manages multicast routing activities, and consists of two sub-components: an intra-domain routing manager and inter-domain routing manager. The intra-domain routing manager maintains policies and algorithms that deal with multicast routing activities within an NGI subnetwork. The inter-domain routing manager is more concerned with functions for handling interoperability between its own subnetwork and other NGI subnetworks that are managed under different policies. Both of these routing managers closely interact with the membership control functional unit in the group management function component. The purpose of their interaction is to establish the multicast routing information that will be conveyed to the routing control functional component in the Network Control functional group, which is where in the NGI function hierarchy that the actual routing algorithm is applied to compute a specific routing tree.

As discussed for routing more generally in Section 3.4, the way routing information is collected is quite different for intra-domain and inter-domain multicast routing managers. The intra-domain routing manager can obtain detailed network information within an NGI subnetwork to help it find optimal routing configurations, whereas the inter-domain routing manager depends on information from other subnetworks that often employ different administrative policies.

The inter-domain routing manager maintains an inter-domain multicast routing protocol in order to interoperate effectively with other subnetworks in the NGI backbone network. Scalability is one of the major issues relating to the design of an inter-domain multicast routing protocol. The scalability of a multicast protocol can be evaluated in terms of how much its overhead grows with the size of the internetwork, the number of groups, the size of each group, and the distribution of group members. That overhead, in turn, consists of the resources that are consumed in the network elements and communication links – resources such as memory space, processing, and communication bandwidth. A number of inter-domain multicast routing protocols have been introduced in recent years, such as Protocol Independent Multicast (PIM) [29], Core Based Trees (CBT) [30], and Distance-Vector Multicast Routing Protocol (DVMRP) [31].

In the NGI functional architecture, the responsibility for both inter-domain routing management and intra-domain routing policies lies with the AutoPops. Thus, AutoPops in the NGI system implement the intra-domain and inter-domain routing protocols on different interfaces, and handle the forwarding of data across domain boundaries of NGI subnetworks. Each AutoPop is also responsible for exporting selected routes out of its own NGI subnetwork and into the higher-level domain that joins it with other domains at its same level in the network hierarchy. More detailed functional description of the AutoPop in relation to multicasting will be given below.

3.6.3. Network Control Functional Group

As in the unicast model, the Network Control functional group for NGI multicast consists of functional components to support the real-time control of network operations, including network control signaling, QoS control, routing control, and reconfiguration control. The functions of each component in this group generally follow the functional definitions given for unicast communication, with the exceptions noted below.

QoS Control

When multicast traffic is included, the QoS control functional component may need to maintain policies for controlling network resources (especially for the IP-domain NGI subnetworks) in order to provide the proper level of QoS. When an ultrahigh-speed WDM network interoperates with lower-bandwidth networks (e.g., IP and ATM), traffic congestion can occur at the boundary nodes (i.e., the AutoPops) that connect the higher- and lower-capacity channels. Congestion at the edge of a subnetwork can introduce

poor bandwidth utilization in other subnetworks that interoperate with it, and can thus lead to performance degradation throughout the entire NGI system.

The problem becomes even more profound when traffic from multiple ultrahigh-speed WDM channels merges into a single lower-speed (non-WDM) channel. This can happen in both the unicast and multicast environment, but it can bring about more serious problems in multicast sessions. In the case of unicast communication, a sender can easily control the message transmission rate with a single receiver so as to minimize buffer overflow and resulting packet loss. This is not so easily accomplished in multicast, especially with large, geographically diverse receiving groups in which the optimal reception rates can vary greatly. When the transmission rate can not be controlled effectively according to the current state of the network (i.e., traffic congestion in this case), there is a greater chance of packet loss due to buffer overflow at the congested routers (or switches). The affected receivers are destined to suffer from message loss, with the end result being poor system performance.

Routing Control:

The routing control functional component is responsible for receiving group and security information and forwarding it to the routing manager in the Multicast Management functional group. Different interfaces for intra-domain and inter-domain routing operations should be maintained by the routing control component.

Reconfiguration Control:

Depending on the multicast routing protocols used in the WDM network (e.g., the WDM Multiple VP Configuration routing method presented in section 6.2 of Reference [25]), the reconfiguration control functional component may need to interact with the Group Management functional component to switch the current configuration of connections into one another.

3.6.4. Information Transport Functional Group

This group consists of switching, interworking, and transmission functional components. Multicasting requires additional functions in each functional components beyond those defined for the unicast environment, and our discussion here will focus on these additional functionalities.

Switching

A key function of switching is data forwarding between two network nodes. In systems like the NGI network in which IP and WDM technologies co-exist, the switching function can be performed in either the electronic or optical domain. Switches operating in electronic domain (e.g., IP routers) can easily support protocol data³ processing for each packet in the link layer. Packet processing offers great flexibility in managing network traffic, including flow and congestion control, routing, and traffic monitoring.

Most of the current WDM switches operate in the optical domain and offer static routing that is feasible for circuit switching with slow reconfigurations. A WDM network with such switching elements functions as a passive communication channel. It provides point-to-point connectivity between a pair of IP routers, and lacks the dynamic routing capability required for implementing dynamic group membership in multicast communication. In other words, in IP-over-WDM multicasting, a WDM network acts as a single-hop, ultra high-speed subnetwork.

Transmission

The data replication function required in multicasting *can* be supported efficiently in the WDM optical domain. Some optical devices, such as the optical coupler and optical power splitter, can implement the *Drop-and-Continue* function in a WDM switch (see Section 3.3), which drops a portion of an incoming

³ A *protocol data unit* in a packet refers to a packet header that is produced and managed by the corresponding protocol.

optical signal and passes the remainder onto an outgoing optical interface. This function can be used as a building block to support packet replications for multicast routing at the WDM layer.

Interworking

As noted, the AutoPop in the NGI backbone network has been defined to interface with heterogeneous subnetworks, acting in this environment as a domain border switch (*or* router). In addition to the functions found in ordinary switches used for unicasting, the NGI AutoPop also includes a set of functions for handling protocol conversions as well as the forwarding and receiving of inter-domain protocol messages.

Because a multicast group may contain members in more than one subnetwork, multicasting complicates the use of subnetworks in general. For example, to minimize the size of the database that each router within a subnetwork must maintain, each router only keeps information about multicast groups that have members in its own subnetwork. To accomplish full connectivity, then, it is the AutoPops (or a *subset* of AutoPops)⁴ that forward group membership information and multicast messages between subnetworks. These “information-forwarding” AutoPops have several key functions:

Since each information-forwarding AutoPop is attached to a subnetwork, it receives the multicast link status reports from that subnetwork and thus knows all of the multicast groups in it. By the same token, because each subnetwork includes at least one inter-domain information-forwarding AutoPop, its multicast group information is known to at least one router (*or* switch) in the backbone network.

Switches (*or* routers) in the backbone network can exchange information about multicast groups so that all switches in the backbone know which subnetworks contain members of each group.

Each information-forwarding AutoPop in a subnetwork can also function as a wild-card multicast receiver. Such AutoPops automatically receive all the multicast messages generated in a subnetwork, regardless of the destination group. They forward a multicast message if necessary, or discard it otherwise. They also guarantee that all multicast traffic originating in a subnetwork is delivered to information-forwarding AutoPops, and then on to the backbone network if necessary.

4. Network Management Approaches for NGI

We now develop a network management architecture for the NGI backbone that provides the functionality of the Network Management functional group of the overall network architecture.

4.1. NGI Network Management Overview

In this section, we describe the NGI network management environment and how this environment drives the network management approach.

4.1.1. NGI Network Environment

The network environment envisioned for the NGI is that of a large scale network of networks with nationwide geographical scope that spans multiple network administrations. The network portion that is under the control of a network administration is referred to as a *domain*. A domain is made up of components of different technology, such as IP, ATM, SONET, and WDM. Hosts attached to the network contain applications that interact by exchanging multi-media information with varying degrees of Quality of Service (QoS) requirements. It is also envisioned that these applications are capable of adapting to changes in the QoS delivered by the network. An important characteristic of DoD applications is that application interactions are long lived, possibly lasting several days.

To provision end-to-end QoS for applications, the resources in the different underlying communication layers of the network have to be managed effectively taking into account the dependencies among the layers in aspects such as configuration, fault handling, and performance. Some example scenarios requiring integrated management are given below:

⁴ The decision whether or not all AutoPops take part in the information-forwarding process should be made carefully, based on strategic requirements and the availability of the network resources such as memory space and computational power in the AutoPops.

- Provisioning the QoS for an application level information flow (stream) requires that the application stream is transported either by establishing a dedicated IP layer flow between the hosts or by multiplexing the application stream over an existing IP flow. This decision should be made taking into account the application level QoS parameters (such as delay, client application request rate, and server application throughput) and the QoS parameters of the IP flows (such as average traffic rate, peak traffic rate, delay, and average error rate). Towards this, application level QoS parameters have to be mapped (by the middleware) to IP layer QoS parameters. After the application stream has been setup, the middleware has to continuously monitor the QoS delivered to the application, and alert the application in case the QoS guarantees are violated so that the application may take appropriate corrective actions.
- Establishment of the IP network configuration using an underlying link layer (such as ATM or WDM) requires that routers and hosts are connected with each other using ATM or WDM connections with specific QoS requirements on these connections. The IP layer configuration management policies have to determine when these connections have to be setup. Some of them may be set up as permanent connections, while others are set up dynamically based on traffic demands.
- A more complex scenario that requires integrated management across IP and link layers is dynamic reconfiguration of the IP network to adapt to changes in IP layer traffic and failures in the underlying link layer communication.

Integrated network management thus entails that each communication layer perform the following functions:

- It should manage its resources taking into account the QoS requirements of its clients and the QoS capabilities/guarantees of the layer below. This involves QoS mapping, and resource scheduling including multiplexing.
- It should monitor the traffic generated by the client and ensure that the traffic does not violate the QoS level agreed by it and the client. It may either discard the traffic that violates the QoS levels or handle it with a degraded QoS.
- It should monitor the QoS delivered by the layer below. If that QoS degrades, it should adapt the resource allocations in its layer and make these changes transparent to its clients to the maximum extent possible.
- In case it is unable to deliver the level of QoS guaranteed to its client, it should alert the client of this event and await corrective actions by the client.

4.1.2. NGI Network Management Study Goals

The following sections of the report describes a network management architecture and design for managing the network environment described above. The proposed design addresses only a subset of the integrated management capabilities described above. More specifically, the following management capabilities were addressed in this study:

- Integrated QoS management spanning Application and IP layers. Towards this end, after first outlining our communications QoS model in Section 4.2, an Application Layer QoS Model and an IP Layer QoS Model are proposed in Sections 4.3 and 4.4 respectively. Mechanisms realizing these models are also proposed. In the proposed design, QoS components in the middleware in end systems and QoS components in the IP network, including routers and IP network management systems (IPNMSs), cooperate to provision end-to-end QoS for applications, as discussed in Section 4.5. The proposed design supports a multi-domain IP network environment, as discussed in Section 4.6. Mechanisms for ensuring the security of End System – IPNMS, IPNMS- Router, and IPNMS-IPNMS interactions are also proposed.
- Configuration management of IP/WDM networks. A scheme for configuring IP/WDM networks was proposed in [32] as part of this study. This scheme takes into account the limitations of

current-day WDM networks, where wavelength connections are at a premium. Readers are referred to [32] for more details.

- Inter-domain WDM network management. An inter-domain WDM network management architecture supporting configuration, connection, fault management functions was also proposed [33]. A novel aspect of the proposed design is that it enables multiple WDM network administrations to dynamically and cooperatively form a federation to support end-to-end WDM connections spanning the multiple network administrations. Again, readers are referred to [33] for more details.

4.2. Communications QoS Model

As discussed in the introduction to this report, the spectrum of applications envisaged to be supported by the NGI can be categorized into the following two classes: (1) *discrete communication* class and (2) *continuous media communication* class. Examples of the former class include a request-response application model, supporting both unicast and multi-cast interactions and typically exist in a client-server scenario. The latter class of applications comprise continuous data exchange, both unicast and multicast data, and typically exemplify the producer-consumer scenario. We envisage that at the highest level, most of the data streams generated and used by applications will fall into one of these two classes.

Applications belonging to each of the above two classes may in turn require varying degrees of service guarantees (which we refer to as Quality of Service (QoS) assurances), from the underlying networks. Based on the QoS requirements, the applications in each of the above mentioned classes can again be categorized as: (a) those that do not require explicit QoS guarantees, and (b) those that require explicit QoS guarantees. Examples of category (a) typically consist of applications that receive best-effort service, e.g., web browsing with no explicit QoS assurances and unrestricted network access. Category (b) in turn is composed of application sets that are: (i) largely delay sensitive (e.g., voice and real-time video), (ii) largely loss sensitive (data, non-real time video images) and (iii) a mix of (i) and (ii) e.g., GCC, distributed multimedia.

The applications that require explicit QoS guarantees, be they of the discrete communication class or continuous media communication class, are typically characterized by long session holding times, though it is possible for certain high-priority short duration sessions to request QoS assurances⁵. The QoS assurances, namely delay and loss guarantees typically translate into timing and throughput constraints that are required to be met at the application layer. In addition to the throughput and timing constraints, certain application streams may allow for a splitting of a composite stream into different streams (e.g., a multi-media application composed of voice, video and data may allow the voice and video parts of the composite stream to be transmitted as one stream and the data part of the composite stream to be transmitted as another separate stream through the network). In such a case, the application streams may require additional synchronization guarantees between their individually transmitted streams. We address this issue also in our work. Finally, certain applications may also require security from the underlying network as part of their service assurance. In this work, we focus on the throughput, timing and synchronization issues and leave the security considerations for future work.

Some related work in this area includes:

- *QuAL (Columbia)*: In this work, the authors propose a new programming language called Quality Assurance Language (QuAL), that can be used to specify and manage QoS communication constraints at the network and application levels[34]. QuAL provides high-level abstractions that the processes at the application layer can use to communicate their QoS requirements to an underlying network. The focus is on distributed multi-media applications.
- *Comet Group (Columbia)*: A variety of work in the area of Open Programmable Multimedia Networks with emphasis on QoS in multiservice networks is being performed by the Comet Group

⁵ There also does exist the possibility of long duration sessions that do not require explicit QoS guarantees, though for the most part, such applications will be characterized by a very low priority.

at Columbia[<http://comet.ctr.columbia.edu>]. In particular, some of the related work are as follows: (a) the work on multicast and multimedia service management using CORBA-based techniques[35] and (b) End-to-End QoS Mapping in Multimedia Networks[36,37]. In (a), the authors propose a generic object model which includes a set of cooperating objects that can be used to capture the interaction between the service delivery and the service management system. In (b), the authors propose a QoS mapping technique between application specified PDUs and network PDUs and study the performance of their mapping rule for an ATM network subjected to video-clips (using JPEG and MPEG).

- *QuO (BBN):* In this work, the authors propose the use of Quality of Service for Objects (QuO) that can be used with CORBA-based systems to perform QoS translation and management between the application and network layers [38]. Use of a QoS Description Language (QDL) is proposed. In essence, the applications use QDL to specify their QoS requirements at a high-level. The necessary translation to network layer QoS and management of the QoS parameters is achieved with the proposed QuO. Elegant techniques are proposed by the authors in this context.
- *Quality of Service Architecture Project (Lancaster University, U.K.):* In this work, the authors propose a framework for the specification and implementation of QoS properties of multimedia applications over ATM networks [39]. The framework incorporates notions of flow, service contract, and flow management. A flow is a unidirectional application stream that may be unicast or multicast. A service contract represents the QoS agreements between an application and the underlying transport layer with regard to a flow. It contains the value of each QoS parameter specified by the application, the kind of guarantee (deterministic, statistical, or best-effort) required for each QoS parameter, the actions the transport layer should take when it violates the guarantees specified in the contract (notify the user, terminate the flow, or offer a degraded QoS), and information concerning the duration of the flow (the start time and the end time).

In all of the above mentioned related work, we would like to point out that one of the most important difference between our work with the others cited above is that we make a distinction between continuous and discrete communication media, while the prior work has focused only on one of the two classes of applications. In addition, we propose and focus on an IP QoS model that uses the evolving differentiated services paradigm and derive detailed solution techniques for such a model, which forms a novel aspect of our work. While the prior work does not mention of any IP QoS model, we observe that the underlying IP layer QoS model plays a very important role in the overall performance and management. Further, the focus of the other related work is mainly multimedia, whereas we give an equal weight to other possible applications that can exist within the NGI application suite.

4.3. Proposed Application Layer QoS Model

This section is devoted to a discussion of the proposed application layer QoS model for the application suite discussed in Section 4.1. In the discussion below, we propose two generic QoS models, one for the discrete communication class of applications and another for the continuous communication class of application suite. For each of these models, we have identified a canonical specification format. These formats are discussed next.

4.3.1. Continuous Communication Class of Applications

Applications in this class are characterized by a continuous flow of data that is generated and consumed by end-users at a certain negotiated rate and size. Information is sent in units called frames, and thus traffic in this class is characterized by a certain frame rate and frame size. In light of the main traffic characteristics, the proposed application layer QoS model requires the users of this class to specify the following: *average frame rate (r)* , *peak frame rate (p)* , *minimum frame size (m)* , *maximum frame size (M)* , *maximum burst size (b)*. We use two rates, namely, the average and peak frame rate to capture both steady (e.g., CBR) type and bursty (e.g., VBR) type of applications. In the steady traffic (CBR) case, the average and peak rates are the same, and hence the application need specify only one rate. In the bursty traffic (VBR) case, the average and peak frame generation rates will differ. For such traffic, the frame sizes may also differ, hence,

the application can specify the different frame sizes by merely providing limits on the sizes it generates, i.e., a minimum and maximum frame size. Additionally, since bursty traffic may be generated in “clumps”, a burst size that represents the maximum number of frames that can be generated consecutively, is also required by the application. Together, these rates and sizes may be used by the network layer to provide appropriate resources in order to guarantee the requested QoS by each of these applications.

In addition to the frame rate and size, we observe that the relative timing between the generated and consumed frames must be maintained. This leads to the following additional parameters that the application need to specify, namely, *maximum frame delay (D)* and *minimum frame delay (d)*. These frame delays indicate, respectively, the maximum and minimum amount of time that may elapse between displaying the consecutive frames in the continuous data stream being transported by the network. In order to prevent overburdening the intermediate network routers (since the routers will serve as a multiplexing point for a host of application streams), we require that the routers be sensitive only to the maximum frame delay, i.e., a frame within a network should not be delayed by a certain “delay budget” computed for that particular node⁶. However, both the maximum and minimum frame delay information are crucial to and are utilized by the end-system middleware which is instrumental in buffering the information received from the underlying network and channeling it to the appropriate application. Thus for example, since the intermediate routers are not sensitive to the minimum frame delay, it is the onus of the middleware to buffer frames that arrive before their scheduled time and hand them over to the application at the specified minimum frame delay interval. Since the middleware at an end-system handles only those subset of applications that are directed to it, it should be able to handle the inter-frame dependencies (both min and max delays) without much problem.

Since different continuous media streams may be sensitive to different amounts of information loss, the application should also specify its information loss sensitivity via the parameter *loss tolerance(l)*, so that this information may be used to appropriately handle situations such as: (a) loss of data within the network, and (b) congestion.

Next, recall, that certain multimedia application streams can transmit their voice/video and data on two different streams (s_1, s_2), if they chose to do so. To allow for such a possibility, we require the corresponding applications to specify the interdependencies between the two individual streams, in addition to the parameters identified above. The *inter-stream timing (ist)* dependency can be specified as: $| \text{Delay}(s_1) - \text{Delay}(s_2) | < k * D$, where $\text{Delay}(s_1)$ and $\text{Delay}(s_2)$ are the delays experienced by streams s_1 and s_2 , D is the maximum frame delay specified, k is an integer greater than zero, and $|a-b|$ is used to denote the absolute value of the difference between variables a and b .

Finally, the applications may also choose to optionally specify a value for the maximum time they are willing to wait if communication within the network were to be interrupted due a network facility failure (e.g., link and/or switch failure) before they decide to abandon the particular continuous data stream, if they think such a specification is pertinent to their communications. We refer to this as the *failure recovery time (ft)*. The underlying IP layer QoS model will then utilize this information in selecting appropriate restoration mechanisms. However, if the applications do not have access to such information, or, are unwilling to specify this information, then the network layer QoS model will employ a set of pre-computed restoration time objectives based on the application class and priority, to restore the communication streams affected by the failure event.

In light of the fact that the QoS specification by the application should be in a language that is easily understood and generated by the applications, we suggest the following units be used by the applications while specifying their parameters in order to obtain their QoS guarantees. Table 4-1 lists the application layer QoS parameters and their units for the continuous communication class.

⁶ Such a per-network node delay budget is obtained from the total end-to-end delay, i.e., from the maximum frame delay specification, and the number of hops traversed enroute the destination node.

Parameters	Units
Avg. frame rate	frames/sec
Max. frame rate	frames/sec
Min. frame size	bytes
Max. frame size	bytes
Frame burst size	bytes
Max. frame delay	seconds
Min. frame delay	seconds
Loss tolerance	frames/sec
Inter stream timing+	seconds
Failure recovery time*	Seconds

* indicates optional parameter
+ indicates a composite stream parameter specification

Table 4-1 – Application Layer QoS Parameters and Units

4.3.2. Discrete Communication Class of Applications

Applications in this class are characterized by discrete request-response type of communication streams. The data streams typically follow a “request-from-client” and “response-by-server” pattern. Examples of such applications include database queries, name-server/directory lookup services, RPC requests, to name a few. In light of the main traffic characteristics, the proposed application layer QoS model requires the client applications of this class to specify the following: *average request rate (r)*, *peak request rate (p)*, *average request size (m)* and *maximum request size (M)*. As with the continuous communication class of applications, we use two rates in the discrete application class case. They are the average and peak request rate, and are designed to capture both a uniform request pattern and a sporadic request pattern, respectively. (Of course, as before, if the request pattern is entirely uniform, then the average and peak request rates default to the same value.) Similarly, the average and maximum request sizes aim at capturing the varying request sizes (where the sizes depend on the details included in the request messages by the respective clients). Again, as before, if the requests are pretty much uniform in size, then the average and maximum request sizes default to the same value.

In addition to the above parameters, the client applications may also optionally specify an *average anticipated response delay (d)* and a *maximum anticipated response delay (D)* if these parameters are important in their transaction. In case they do not specify the above mentioned delay parameters, then the clients are generally satisfied with the delay parameters declared by their servers. Observe however that since the communication pattern is in the form of discrete request-response mode, the parameters “maximum burst size” and “loss tolerance” as defined for the continuous communication class, that represent, respectively, “clumps” of arrivals and the number of frames on an average that can be dropped while still maintaining data coherency, are no longer pertinent to the discrete communication model, and hence not used in its QoS model.

Finally, as with the continuous communication class of applications, the applications in the discrete communication class may also choose to specify (optionally) a failure recovery time, i.e., the maximum time they are willing to wait if communications within the network were to be interrupted due to a failure event (such network facility failures), if they think such a specification is pertinent to their communications. In case they do not specify the failure recovery time, then as described before, the

network layer will employ pre-determined restoration time objectives based on certain guidelines (such as priority, application class, etc.) to restore the data streams affected by the network facility failure.

The application layer QoS model for the server applications, requires a similar set of parameter specifications as outlined for the client applications in the request-response transaction, with the “requests” being replaced by “acceptance”. Thus, we have on the server side the following: the *maximum acceptance rate (p)*, *average acceptance rate (r)*, *average response size (m)* and *maximum response size (M)*. However, while the clients need to specify the anticipated response delays (both average and maximum) only optionally, the servers are required to specify the *maximum service delay (D)* and *average service delay (d)* that the server can provide⁷.

Regarding the units in which the QoS parameters need to be specified, we once again keep in mind the need for an easy-to-specify set of units that the applications can relate to without much difficulty. Hence the units for the client query rates (both max and avg.) are in requests/sec, while the units for the server response rates (both max and avg.) are in replies/sec. The maximum and average request and response size are specified in bytes. The delays (both max and avg.) are specified in seconds.

Table 4-2 summarizes the proposed QoS model for the NGI application suite.

It should be noted that in the case of the continuous communication class, the proposed QoS model is employed to specify the QoS requirements of a communication end point of an application. Hence, the QoS model is applicable both for unicast and multicast application sets.

Application Class	Parameters	Units
Continuous Communication Class (Producer/Consumer)	Rate: Avg, Peak	frames/sec
	Size: Max, Min, Burst	bytes
	Delay: Max, Min	seconds
	Loss Tolerance	frames/sec
	Inter-Stream Timing+	seconds
	Failure Recovery Time*	seconds
Discrete Communications Class (Client)	Rate: Max, Avg	requests/sec
	Size: Max, Avg	bytes
	Anticipated response delay: Max, Avg*	seconds
	Failure recovery time*	seconds
Discrete Communications Class (Server)	Rate: Max, Avg	replies/sec
	Size: Max, Avg	bytes
	Service delay: Max, Avg	seconds
	Failure recovery time*	seconds

* indicates optional parameter
+ indicates a composite stream parameter specification

Table 4-2 – QoS Model for NGI Application Suite

⁷ The maximum and minimum values (for the acceptance rate, response size and response delays) can be obtained either empirically from a histogram of previous and anticipated requests, or by applying appropriate stochastic models for the types of queries/requests. i.e., by assuming a certain probability distributions for them based on observed data.

4.4. IP Layer Communication/QoS Model

4.4.1. Review of Existing and Emerging Models

The Internet was originally designed to provide best effort service, i.e., a delivery mechanism that does not guarantee any loss or delay values to the applications using it. Instead, the loss and delay characteristics of the applications depend on the instantaneous load on the network. While such a strategy works well under lightly loaded conditions, the performance will deteriorate steeply under moderate-to-highly loaded conditions. Furthermore, there are many applications that require varying degrees of performance guarantees, thereby requiring the underlying network to provide different degrees of QoS. This has led to the design and development of “suitable” QoS provisioning models for the Internet. We will provide a brief overview of some of these models in this subsection.

The RSVP and Integrated Services Approach to QoS Provisioning in IP Networks:

One approach to providing QoS in IP-based networks is to use the RSVP (Resource ReSerVation Protocol) signaling mechanism to negotiate and allocate resources based on an application’s requested QoS from the network [17,40,]. The QoS guarantees can be either statistical (e.g., via Controlled Load Service) or deterministic (e.g., via Guaranteed Service). While allowing applications to reserve resources in order to provide QoS guarantees, RSVP relies on two key concepts: (a) flows and (b) reservations. Flows are used to embed the notion of a connection in the inherently connection-less IP environment. Flows are traffic streams from a sender to one or more receivers. Thus at a high level, flows are representative of sessions. RSVP identifies a flow by its destination IP address and, optionally, a destination port. In addition, RSVP may also use a particular source IP address or source port, or use the *flow label* field in the IP basic header together with the source address in its identification of a flow.

RSVP uses *flowspecs* to specify the traffic characteristics for a given flow. RSVP however, does not itself understand this flowspec, but simply acts as the vehicle to pass the flowspec from the application to hosts and routers along the flow’s path. Flowspecs specify the desired QoS which can be used to set appropriate parameters in a node’s packet scheduler or other link layer mechanisms in order to enforce and realize the requested QoS. A flowspec is composed of two components: (a) Tspec (T for “traffic”) and (b) Rspec (R for “reserve”). They contain bandwidth and delay specifications, and together, they are used to negotiate and reserve resources in the network. RSVP supports both unicast and multicast flows.

Reservation of resources in RSVP is initiated by the traffic receivers. Receiver initiated reservations gives the protocol a great deal of flexibility for handling multicast flows, particularly those that are diverse or dynamic. Note that while RSVP is a specific mechanism that can be used to reserve resources based on a certain QoS requirement, it does not dictate or prescribe the use of a specific routing engine to obtain the routes that satisfy the resource requests. A commonly used routing scheme in the Internet however is the OSPF (Open Shortest Path First) strategy, which is a dynamic link-state routing protocol.

Despite its flexibility to provide QoS in IP-based networks, RSVP suffers from a major drawback, i.e., it does not offer graceful scalability to large sized networks. This is because a basic tenet of RSVP is to allow each individual flow to specify distinct QoS attributes which are required to be maintained at each of the intermediate routers as well as the end systems. Thus it becomes necessary to handle the bandwidth and buffering resources at every router for each and every flow in the network. In addition, control traffic (PATH and RESV) messages need be continually exchanged for each of these flows during their lifetime, in order to maintain the requested resources (and thereby prevent the system from defaulting to a “best-effort” service mode). This obviously leads to scaling concerns when large numbers of flows need to be supported.

The ST2 Protocol Approach to Providing QoS Engineering in IP Networks

Another approach to QoS engineering in IP-based networks is to allocate resources based on negotiated QoS parameters via the ST2 protocol[41,42]. ST2 is a connection oriented internetworking protocol developed to provide QoS to applications that require QoS guarantees from an underlying IP network. It can be used to provide reservations for streams across network routes, thereby guaranteeing a well-defined QoS to such streams.

The ST2 protocol consists of two protocols: (a) ST and (b) SCMP (Stream Control Message Protocol). The ST part is used for the data transport and is fairly simple so as to achieve fast data forwarding and low communication delays. The SCMP protocol is used for all control-related functions and is relatively more complex.

ST2 defines two QoS classes: (a) QoS Predictive and (b) QoS Guaranteed. The QoS Predictive class implies that the negotiated QoS may be violated for short time intervals during the data transfer. (In spirit, it can be viewed as the counterpart of the Controlled Load Service Class of the Integrated Services Model discussed earlier). An application has to provide values that take into account the “normal” case, e.g., the “desired” message rate is the allowed rate for the transmission. Reservations are done for the “normal” case as opposed to the peak case. Support of this class is required by all ST2 implementations.

The QoS Guaranteed class on the other hand, requires that the negotiated QoS for the stream is never violated during the data transfer. The application has to provide values that account for the worst possible case. As a result, sufficient resources to handle the application are reserved. Note that this strategy may lead to overbooking of resources, but it is capable of providing strict real-time guarantees. (In spirit, this class can be viewed as the Guaranteed Service Class of the Integrated Services Model discussed earlier.)

Similar to RSVP, ST2 also uses flowspecs to describe the required characteristics and the parameters included in the ST2 flowspec are bandwidth and delay specifications. However, the details of the parameters for ST2’s flowspec and RSVP’s flowspec differ. ST2 uses CONNECT, ACCEPT and REFUSE messages during the connection setup phase to either accept or refuse connections. Explicit tear down messages need be sent to mark the end of a session in ST2. A local resource manager works in conjunction with an ST2 agent in each router to check if resources are available to match the flowspec. ST2 can support multicasting.

An example network that uses ST2 is the German PTT which uses ST2 as its core protocol in its BERKOM MMTS project to provide multimedia teleservices such as conferencing and mailing. Another example is the work in [43] that proposes the use of ST2 in its QuAL prototype.

However, a significant drawback of ST2 (like RSVP) is its ability to scale elegantly to large sized networks. By requiring the individual routers to have local resource managers for each of the ST2 streams (similar to RSVP), the ability to scale gracefully to large sized networks is severely compromised.

The Differentiated Services Approach to Service Allocation and QoS Engineering in IP Networks

The evolving Differentiated Services Model is yet another mechanism that can be used to allocate the bandwidth of the Internet to different users based on their specific requirements and thereby provide varying degrees of QoS guarantees to the applications using the Internet[15,19]. The differentiated services mechanism is aimed at providing users with a predictable expectation of what service the Internet will provide to them in times of congestion, and is capable of accommodating different users with different levels of service requirement in the network.

The underlying principle of the differentiated services mechanism is as follows. Each user defines a certain “service profile” to specify their traffic attributes and mark their traffic as either “in” or “out” profile, based on their declared traffic characteristics. The routers use this service profile information for the following: (a) to reserve resources for the “in” profile packets, and (b) to preferentially drop traffic that is tagged as

being “out” in case of a congestion. Thus, at the routers inside the network, there is no separation of traffic from different users into different flows or queues. Instead, all the user packets are aggregated. However, since the various users can have different service profiles and therefore negotiate resources based on their own profile during the admission phase, this results in different quantities of “in” and “out” packets for the various streams using the IP network. This in turn results in different amounts of “in” packets being served for the different applications, based on their individually negotiated QoS.

To allow for a great deal of flexibility, the differentiated services mechanism allows for both “sender driven” and “receiver driven” negotiation of service profiles. The service profile, as its name indicates, is used to describe the individual application’s traffic characteristics. Included in it can therefore be both bandwidth and delay requirements. Profile meters are used to ensure that the service profiles declared by the applications are not violated.

Regarding the degree of assurance that a user can expect with this mechanism, we have the following. In light of the fact that users may want both probabilistic as well as deterministic QoS guarantees from the underlying network (depending on how much they want to pay), the differentiated services model can be used to provide both statistical bandwidth allocation as well as fixed/guaranteed bandwidth allocation. In the case of statistical bandwidth allocation, the user receives certain “expected capacities”, i.e., in steady state, the user will obtain a certain minimum capacity from the underlying network. However, the instantaneous bandwidth will be uncertain. This therefore leads to providing statistical guarantees to the user and is usually purchased by users that are less demanding i.e., they purchase a service that is “usually” available but may still fail with a low probability.

On the other hand in the case of fixed/guaranteed bandwidth allocation, specific network resources are identified and reserved and the user is assured of adequate bandwidth during the entire duration of their session. This allows the network to provide certain deterministic guarantees on the delay, for example. Users requiring this service are often required to pay more and will in turn be assured of their requested performance at all times, i.e., there will be no uncertainty about resource availability to these applications.

4.4.2. Proposed IP Layer Communication/QoS Model

Based on our review of the existing and emerging models for QoS provisioning at the IP layer, we observe that the evolving differentiated services approach provides the dual advantage of: (a) adequate flexibility to guarantee varying degrees of QoS, and (b) scaling gracefully to large sized IP networks. We therefore propose a QoS Model for the IP layer based on the differentiated services approach as also in [17].

The details of the proposed model are as follows. As discussed in Section 2.1, QoS sensitive applications can be broadly classified as: (a) largely delay-sensitive applications that require stringent real-time guarantees and are relatively tolerant to random loss (e.g., voice and real-time video), and (b) largely loss-sensitive applications that do not require stringent real-time guarantees but are relatively sensitive to random losses (e.g., data).⁸

At the IP layer, we therefore provide two distinct classes to support the different QoS requirements of the various applications. The first class, also referred to as Class 1, caters to traffic that are largely delay sensitive and hence require stringent delay requirements. The second class, also referred to as Class 2, caters to traffic that are: (i) non-delay sensitive (i.e., largely loss-sensitive), and (ii) neither loss nor delay sensitive, i.e., QoS insensitive applications which just subscribe for “best-effort” services (e.g., web browsing and other short duration applications which do not ask for any explicit QoS guarantees). The class distinction is done based on a user-specified “service profile”. The service profile is in turn used by a profile meter at the network ingress point to mark the user traffic as either “in” or “out” profile. (Since one possible place to implement the profile metering is at the middleware level, we will discuss the details of the profile meter below.) The intermediate routers and the end subsystem use the “in” and “out” profile

⁸ Of course, there can also be applications that are a mix of categories (a) and (b). We will show how such a mixture of applications are handled in the proposed IP layer QoS model.

markings to differentiate and hence preferentially discard the “out” packets during periods of network congestion or unanticipated network behavior⁹.

Typically, the suite of applications that subscribe to Class 1 service are required to provide deterministic characterization of their input traffic (in order to be able to get the deterministic guarantee promised by Class 1). This in turn implies that their traffic behavior should be of predictable nature, an example of which is CBR traffic. In case of bursty traffic that need deterministic guarantees, we propose that a leaky-bucket type of shaping mechanism (with an appropriate bucket size and token rate) be used in conjunction with a weighted fair-queueing mechanism. While it is possible to realize fixed delays in the latter case, we observe that such a possibility will have to be made at the expense of increased resource allocation to such applications, hence trading network utilization/efficiency for such a guarantee.

On the other hand, the suite of applications that subscribe to Class 2 service are often characterized by uncertainty in their traffic behavior, e.g., VBR traffic. The guarantees provided to such a class are therefore statistical. Most often, the applications that subscribe to this class are tolerant to delay but relatively sensitive to loss. Hence, to such a class, guarantees about a certain minimum bandwidth can be made by reserving certain amounts of resources based on the “in” profile traffic for such a class, thereby making it possible to guarantee a minimum loss value for the loss-sensitive applications. However, given the bursty nature of the input, this class of applications is characterized by a potentially significant amount of “out” profile traffic. Thus only statistical guarantees can be made about the resource availability (and hence loss probabilities) to such a class.

Having discussed the general traffic characteristics of the two classes, we next describe how the above proposed IP layer QoS model can be implemented to realize the promised performance from the two classes for the various application types. Firstly, all of the incoming traffic injected at the IP layer has to follow a given service profile that is understood by the IP layer. We propose the use of a format that corresponds to the Token bucket algorithm, i.e., the user’s traffic is presented to the IP layer (either directly by the user or after a format conversion by the middleware) in the following format: $TF = f(r,p; b, M, m)$ where TF stands for traffic format, $f(\cdot)$ represents a function of the arguments within the parenthesis, and “ r ” and “ p ” denote, respectively, the average and peak rates in bytes per second. “ b ” denotes the maximum burst size, i.e., the maximum number of packets that can consecutively arrive at the peak rate “ p ”. “ M ” and “ m ” denote, respectively, the maximum and minimum packet sizes measured in bytes. The significance of the latter (m) is that since it is the minimum policed unit, all packets with fewer than “ m ” bytes are treated as having “ m ” bytes, for profile metering.

Thus the output of the profile meter is a sequence of bytes which are stored inside the network in two queues, namely the Class 1 queue and Class 2 queue, based on the application’s requested QoS guarantees. The traffic emerging from the profile meter is also marked as “in” or “out” profile by the profile meter, based on the particular application’s service profile and negotiated resource requirements. In the case of Class 1 traffic, we only admit the “in” profile packets into the Class 1 queue within the network layers. For Class 2 traffic, both “in” and “out” profile traffic are admitted to the Class 2 queue.

The philosophy for such an admission policy is as follows. In the case of Class 1, since deterministic guarantees are provided, the network has to have a precise knowledge of the incoming (and hence negotiated) traffic because resources have to be precisely allocated for this class. By admitting only the negotiated traffic with a deterministic pattern, the routers can allocate resources (both processing power and bandwidth) precisely to the various applications, thereby guaranteed a fairly precise performance at each router stage. Since the end-to-end delay is a superposition of the intermediate delays, achieving fixed delays at each of the intermediate stage helps realize a fixed end-to-end delay, as guaranteed by such a service class. In order to do so, no uncertainty of traffic should be present at the various routers, which is therefore achieved by admitting only the “in” profile packets into the network.

⁹ The intermediate routers may also have their own profile meters to monitor the aggregate traffic flowing through them.

In the case of Class 2 service, since the guarantees are statistical in nature this in turn implies a certain “expected capacity” with a guarantee on the “minimum” capacity is accorded to traffic that use this service class. Thus, the total instantaneous bandwidth is unpredictable in such a case, though a certain minimum steady-state bandwidth (and hence steady-state performance such as average loss probabilities) is guaranteed. Thus, in this case, both “in” and “out” traffic may be admitted, with the “out” traffic being discarded in case of a congestion. This ensures a certain minimum bandwidth (for all of the “in” traffic), but due to the bursty nature of the input traffic, traffic over and above this minimum can still be provided service, if capacity exists. The intent here is that even well-behaved Class 2 streams are prone to injecting non-compliant packets occasionally into the network due to the high degree of traffic unpredictability at the application layer. Such packets should therefore still receive service from the network, so long as there is no congestion in the network. Of course, in case of congestion, the “out” profile packets, which are essentially “non-conforming” packets are promptly discarded.

Regarding service scheduling between and within the two classes, we do the following. First, we observe that QoS provisioning requires a prioritized scheduling mechanism at the routers (as opposed to the FIFO discipline employed in today’s best-effort service Internet). We have two priority queues within each router, one queue corresponding to Class 1 traffic and another corresponding to Class 2 traffic. Class 1 packets are assigned the highest service priority and are hence always served before packets belonging to the Class 2 queue (with the exception that a packet already in service is not pre-empted). Figure 4-1 depicts the proposed service scheduling policy.

As a direct consequence of the above mechanism, Class 1 packets have the entire capacity of the router processor and outgoing link at their disposal irrespective of the load due to the other class. However, the determinism of Class 1 traffic enables a tight control over the actual processor and bandwidth capacities actually consumed by class 1 traffic. (Note this is a direct consequence of admitting only the conforming or “in” profile packets and discarding all of the non-conforming or “out” profile packets into the network.) In particular, it is now possible to predict with fair certainty the exact impact of each new class 1 session, in conjunction with the other ongoing Class 1 sessions on the network resources, which in turn leads to a considerable simplification of the Class 1 admission control procedures. Since there is only one type of packets in the Class 1 queue, the question of priority scheduling within the Class 1 queue does not arise.

In the case of Class 2 service scheduling, we assign to Class 2 packets a lower priority than Class 1 packets. This implies that if a Class 1 packet is present, it will be served in preference to a Class 2 packet. However, in the absence of Class 1 packets, the Class 2 packets will have full service access. In other words, Class 2 packets will have access to the full capacity at each stage minus the deterministic segment assigned to Class 1 streams. Since Class 2 packets may be either “in” or “out” profile, the buffering and service policy within Class 2 is as follows. As long as sufficient buffer space exists, Class 2 “out” packets will be admitted and served. However, should congestion occur, the “out” packets within the Class 2 queue will be discarded to make room for the “in” packets (which recall have a certain minimum loss guarantee). Finally as before, service is non-pre-emptive, i.e., should an “in” profile Class 2 packet appear while an “out” profile Class 2 packet is being served and the Class 2 queue be full, the currently served “out” packet will not be pre-empted from service.

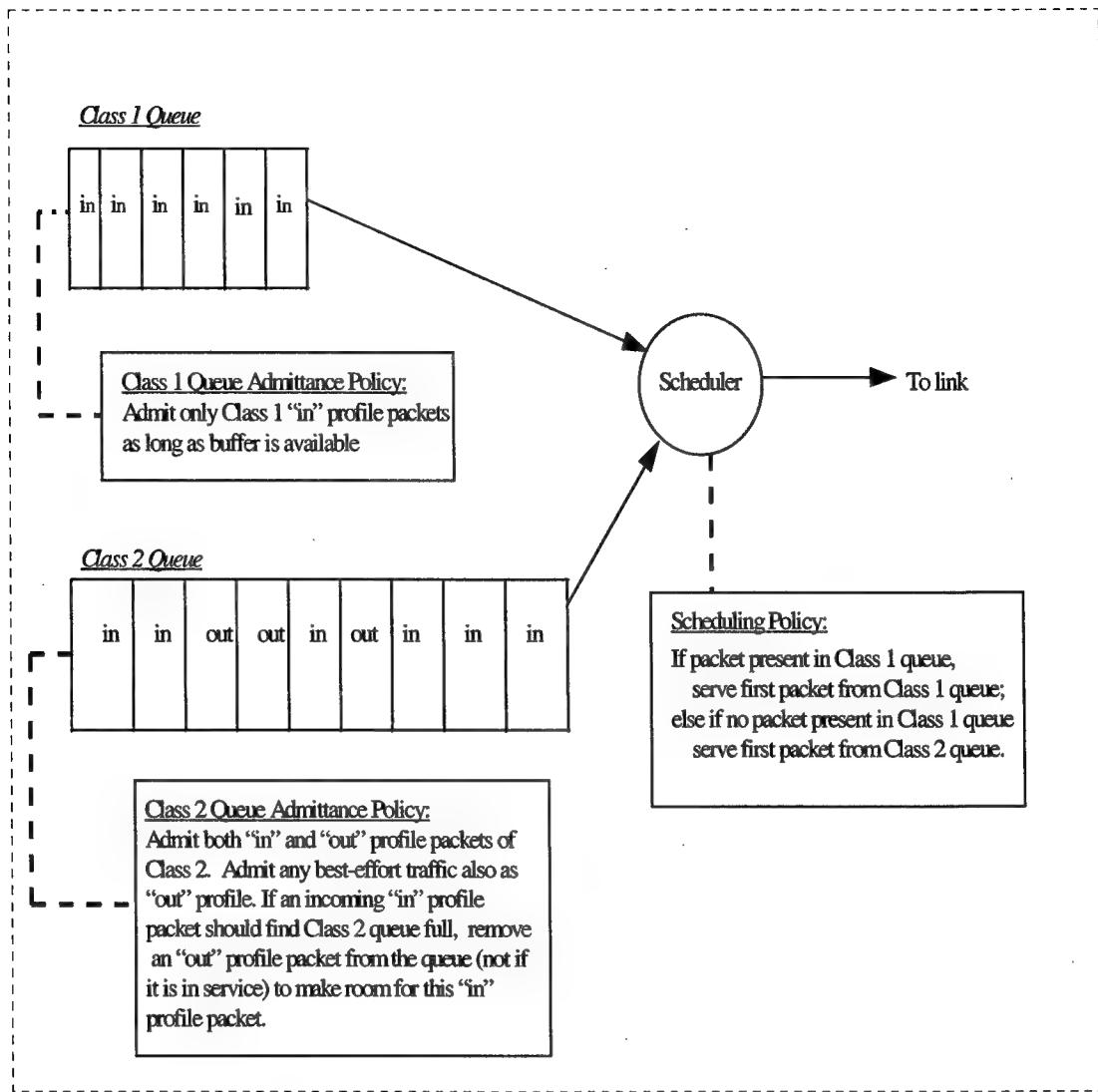


Figure 4-1 – Proposed Scheduling Policy

Recall that we mention that QoS insensitive traffic (best-effort traffic) is also treated as Class 2 in the proposed IP layer QoS model. In such a case, we mark all of the traffic corresponding to such a stream as "out" profile, and admit them into the network. Note, we also do not perform any profile metering of such traffic, since they do not need to negotiate any resources.

Finally, we describe how we handle applications that are a mix of both delay and loss sensitive applications. There are two ways to handle such applications: (1) transport the composite stream as entirely one class, or (2) transport the voice and video components of the stream as Class 1 and the data component as Class 2. In what follows, we briefly discuss each of the approaches.

Approach (1) has two alternatives in turn. One alternative would be to support the composite stream as Class 1 and another would be support the composite stream as Class 2. In case of the first option, the "r" parameter could be set to the "p" parameter provided it is known. In such a case, the composite stream will not only receive delay guarantees, but loss guarantees as well. However, this approach can lead to inefficient utilization of resources. In case of the second option, the composite stream could be sent entirely

as Class 2 traffic. While this can save bandwidth (and hence lead to better resource utilization), the real-time components (voice and video) would only be able to receive statistical guarantees on their requested QoS.

The second approach tries to make good use of network resources without compromising much on the QoS performance, i.e., tries to combine the merits of both alternatives of approach (1) by splitting the real-time components of the composite stream (typically voice and video) and transporting them as Class 1 traffic, and transporting the non-real time components (typically the data component) as Class 2 traffic. Thus, network resources are used efficiently, while still guaranteeing the real and non-real time components their respective QoS. This approach however will require suitable synchronization mechanisms at the receiving sub-system, since the two individual streams will have some relative timing requirements to be met. In practice, if the synchronism between the two streams is only up to a certain limited degree of granularity, then the gigabit/terabit per second platforms that the NGI is expected to support will render this synchronization issue fairly simple. For example, in a large class of multi-media applications (e.g., distance learning), or in a GCC scenario, it might suffice if the synchronism between the voice and video streams and the data stream that supports a shared “whiteboard display” are maintained within a few tens of milliseconds. Timing granularities of this order can be easily provided by the NGI’s gigabit/terabit networks. The savings in network resources (i.e., increase in network utilization/efficiency) coupled with the acceptable performance of this approach, renders it an attractive option to transport such a mix of applications.

4.5. QoS Management Architecture

4.5.1. Intradomain QoS Management Software Architecture

Intradomain Management Architecture

The Intradomain management architecture consists of the following management system components. See Figure 4-2. In this figure, ES stands for *End System*. ER and CR are used to denote, respectively, an *Edge Router* and a *Core Router*. The box named IPNMS represents the *IP Network Management System*. Solid lines are used to indicate data flow and dashed lines are used to indicate control/management information flow.

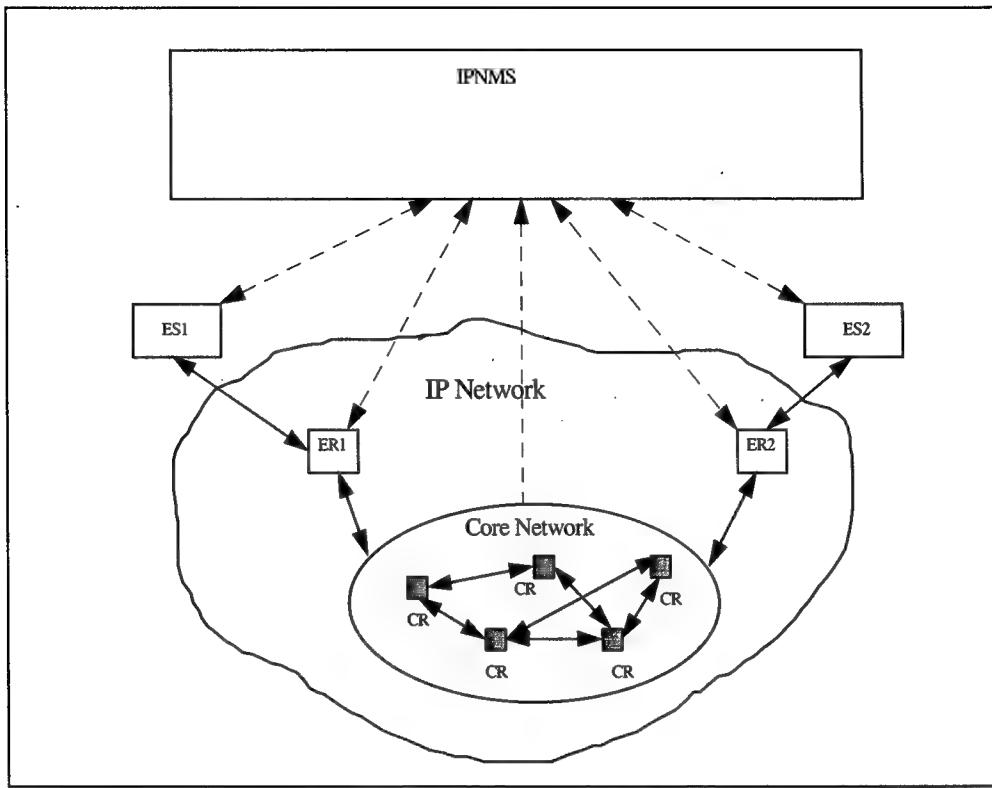


Figure 4-2 – Intradomain Network Management Architecture Components

In the proposed architecture, the IP network is composed of two types of routers: edge routers that provide connectivity to end systems, and core routers that connect routers. To support scalability, complex QoS functions are pushed to the edge of the network while efficient forwarding coupled with minimal service differentiation activity occurs inside the network.

The function of each management system component is described below:

- The middleware in the end systems manage streams. A stream is a unidirectional flow of information (formatted in application defined frames) between applications that are different in end systems. Each stream is characterized by the following set of QoS parameters: bandwidth (peak and/or average), delay, error loss rate, and frame size (maximum and average). The middleware provides support for setup, release, and monitoring of streams that take into account the QoS parameters of the streams. Streams are transported over *IP Pipes*. An IP Pipe is a unidirectional flow of packets between transport layer ports in two end systems. Each pipe is characterized by the following set of QoS parameters: service class (Class 1 or Class 2), bandwidth (peak and/or average), and packet size (maximum and average). A stream is transported over either an IP pipe that is dedicated for the stream or an IP pipe that is shared among multiple streams. Policies and mechanisms for multiplexing several streams over one IP pipe are associated with the middleware that sets up the streams. Several pipes can be set up between a pair of end systems.

- The IPNMS performs network bandwidth management for IP pipes set up between end systems. When an end system needs an IP pipe to another end system, it requests the IPNMS. The IPNMS, based on the current network topology (including the states of links and routers), the routing protocol used in the routers, available bandwidth on the links, and the QoS parameters of the requested pipe, either accepts or rejects the pipe. This determination is done in the following manner. First, the IPNMS executes the routing algorithm using the current network state image that it has, and selects the route for the requested pipe. Then, it checks if sufficient bandwidth is available for the pipe on each link on the selected route. If the check succeeds, it accepts the pipe request; otherwise, it rejects the pipe setup request. If the IPNMS accepts the pipe request, it sets up *meters* in the edge router on the ingress (source) end of the pipe to police and monitor the traffic sent over the pipe. It also sets up bandwidth and delay *monitors* in both ingress and egress edge routers of the pipe to collect information on the QoS actually delivered for the pipe. Further, it updates the available link bandwidth information. The QoS information concerning a pipe that is collected by the IPNMS can be retrieved by end systems by sending a query to the IPNMS. Note that routers do not participate in any reservation protocol in setting up a pipe. Also note that pipe setup and release operations do not involve the core routers at all. To keep track of the current network topology, routers send to the IPNMS information on network topology changes. The following kinds of events trigger edge and core routers to send topology change reports to the IPNMS: Link failure / recovery, router failure/recovery, link capacity changes, and addition/ removal of links.
- Edge routers provide per pipe QoS monitoring and metering (policing) functions. Metering for a pipe is done on the basis of traffic class of the pipe and the bandwidth of the pipe. Packet scheduling is done on the basis of traffic classes. An edge router destroys the meters and monitors associated with a pipe when the IPNMS notifies the edge router that the pipe is released.
- Core routers are aware of only aggregate class traffic and not individual pipes. QoS support in the core routers is through capacity provisioning and class based packet scheduling. On each link associated with a core router, distinct provisioning levels are set for Class 1 and Class 2 traffic. In a core router, packet scheduling is done on the basis of class priority and the link provisioning level for each traffic class.

The proposed management software architecture has the following merits:

- By not requiring routers to participate in any reservation protocol for setting up pipes, this architecture scales well for large networks that may be required to support thousands of pipes concurrently.
- The decision to admit a pipe rests only with the IPNMS. Only an end system, the IPNMS, and the edge routers need to interact in setting up a pipe. Hence, the pipe setup operation can be implemented in a very efficient manner.
- The core routers have no knowledge of pipes. Core routers are concerned only with fast forwarding of packets taking into account class priorities. Thus, QoS support in the backbone network is through a combination of network provisioning and packet scheduling.
- The architecture makes a clear separation between three levels of QoS mechanisms: fine grain stream level QoS support mechanisms, medium grain pipe level QoS support mechanisms, and coarse grain traffic class level QoS support mechanisms. Middleware in the end systems cooperate and collect stream specific QoS information. The IPNMS interacts with edge routers and collects pipe specific QoS information and delivers such information to the end systems on demand. The pipe specific QoS information collected by the IPNMS can be used by the middleware to adapt its decisions concerning mapping of streams to pipes. The IPNMS also collects traffic class specific QoS information from the edge and core routers. Such

information can be used by network engineers and planners for network reconfiguration and evolution.

The above discussion is a brief overview of the proposed management software architecture. The next sections describe the management components and their interfaces in detail.

Related Work

As mentioned above, the proposed QoS model and the management architecture are well aligned with the Differentiated Services model that is emerging in IETF [17,19]. The Diffserv model as outlined in the IETF draft [17] is gaining momentum, and the proposed IP layer QoS model is based on the differentiated services approach. However, the Diffserv model draft does not discuss any management functions. In our work, we use a model that is similar to the Diffserv model and propose an IPNMS that can work within the Diffserv service model framework. Another IETF draft [19], which is also in line with the Diffserv draft [17], discusses a two-bit differentiated services architecture for the Internet. This draft focuses on service aspects, i.e., the service definitions and a two-bit architecture to support the defined services, and does not discuss much about the management aspects. However, the bandwidth broker architecture discussed in [19] has some relevance to the IPNMS management functions and the IPNMS bandwidth management unit design proposed in this work¹⁰, although it does not have any details of the management aspects and design of the bandwidth broker. Thus, while we use the Diffserv IP layer service model and the concept of maintaining one bandwidth allocation agency per domain in order to align our work with the ongoing work within the IETF community, the proposed IPNMS design as well as the overall IPNMS framework and functional units described in this report, are novel to the current work.

The middleware software structure described above has some similarities to the CORBA based control and management architecture for audio/video streams described in [44]. Some of the components described in Section 4, i.e., *Stream Controller*, *Stream Source*, and *Stream Sink*, have their direct counterparts in the architecture described in [40]. But, there is a significant difference between the two architectures. The architecture described in [40] is not concerned with any specific network transport and does not deal with the integration of stream management components with network transport management components. In contrast, the middleware architecture described in Section 4 has been designed with the explicit goal of integrating the stream management components with the IP layer management components. Thus, the middleware architecture includes components that map stream QoS parameters to IP layer QoS parameters, components that support multiplexing of multiple streams over a single IP pipe, and components that interact with IP layer management components to setup, release, and monitor IP pipes that transport streams.

4.5.2. Subsystem Interfaces

Recall that the proposed management software architecture consists of the following subsystems:

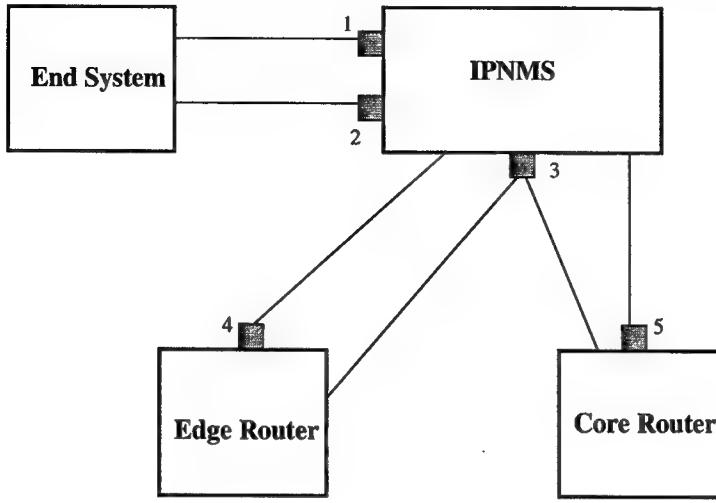
- End System
- IPNMS
- Edge Router
- Core Router

Each subsystem is made up of a collection of objects, each object providing a subset of the management functions associated with the subsystem. This subsection describes the management interfaces for the interactions between these subsystems. In this description, each subsystem is viewed as a black box and the distributed object structure within the subsystem is not considered. The next sections discuss the internal structure of the subsystems. Figure 4-3 shows this high level management software architecture. In this level, each subsystem is viewed as a composite object that supports one or management interfaces. In the

¹⁰ The bandwidth management unit (BMU) is one of the functional units of the proposed IPNMS.

figure, large boxes denote subsystems, small dark boxes denote interfaces, and lines denote client-server relationships. A line connects a client subsystem to a server subsystem. The subsystem that has the interface is the server subsystem, and the subsystem at the other end is a client subsystem.

An overview of the management interfaces illustrated in Figure 4-3 is given below.



1. Pipe Setup/Release Interface
2. Pipe QoS Monitoring/Accounting Interface
3. Topology Change Report Interface
4. Edge Metering/Monitoring Interface
5. Core Capacity Management Interface

Figure 4-3 – Management Interfaces

Pipe Setup/Release Interface

This interface is provided by the IPNMS and is used by the End Systems to setup and release IP pipes. This interface provides operations for setup and release of IP pipes as well as operations for setup and release of monitors for each of the following QoS parameters: bandwidth, delay, and loss. An end system requests the IPNMS to set up an IP pipe by invoking the *setup_pipe* operation. The end system identifies the end points using a *Classifier* structure that contains the host IP address and port number of the source and the destination end points. At pipe setup time, the QoS parameters of the pipe are also specified. Separate operations are provided for setting up monitors for the three QoS parameters mentioned above. Such monitors can be individually setup and released. When a pipe is closed, all monitors are automatically released. Information collected by the monitors are retrieved by the end system using the *Pipe QoS Monitoring/Accounting* interface.

The CORBA IDL specification of the *Pipe_Setup_Release* interface is given in [45].

Pipe QoS Monitoring/Accounting Interface

This interface is provided by the IPNMS and is used by the End Systems to retrieve information concerning actual QoS delivered by the IP network for specific IP pipes. Separate query operations are provided in this interface for querying each of the following QoS parameters: bandwidth, delay, and loss. The information returned for each invocation is the cumulative average statistics for the specific pipe. Three types of average measures are returned: average for “in profile” packets, average for “out profile” packets, and average for all packets. End systems can query the IPNMS either periodically or sporadically. No assumption is made in this interface concerning the polling interval used by the end systems. Recall that a

distinct monitor is associated with each pipe for each QoS parameter, and each such monitor is identified using a unique MonitorId. When it queries the IPNMS, an end system identifies the monitor using the corresponding MonitorId.

An end system that retrieves the QoS information using this interface may use the information either for QoS monitoring or accounting purposes. Hence, this interface is called the *Pipe_QoS_Monitoring_Accounting* interface. The CORBA IDL specification of this interface is given in [45].

Topology Change Report Interface

This interface is provided by the IPNMS and is used by the edge routers and core routers to notify IP network topology changes. The following kinds of events trigger topology change reports: link failure, link recovery, link creation, link deletion, router failure, router recovery, router creation, router deletion, and link capacity changes. Routers communicate such topology changes to the IPNMS by invoking the operation *topology_changed* defined in this interface. The CORBA IDL specification of this interface is given in [45].

Edge Metering/Monitoring Interface

This interface is provided by each edge router and is used by the IPNMS to setup and release meters and monitors for an IP pipe. The IPNMS requests an edge router to setup a meter when the IPNMS accepts a pipe setup request from an end system. Setting up a meter in an edge router triggers initiation of classification and policing functions in the edge router for the associated pipe. The IPNMS requests the meter to be released when it releases the associated pipe. Subsequent to a meter setup for a pipe, the IPNMS may request setup of monitors in the ingress and the egress edge router for the pipe. Separate operations are provided for setting up bandwidth monitor and delay monitor. Such monitors can be individually setup and released.

A bandwidth monitor collects average bandwidth information over a sampling interval that is specified by the IPNMS at monitor setup time. A delay monitor on an ingress router timestamps selected packets, and records these entry timestamps. The IPNMS retrieves these entry timestamps recorded in a sampling interval. A delay monitor on an egress router timestamps received packets that have entry timestamps, and records these exit timestamps. The IPNMS retrieves these exit timestamps recorded in a sampling interval. The IPNMS computes loss statistics for a pipe based on the bandwidth monitor information retrieved from the ingress and the egress router. The IPNMS computes delay statistics for a pipe based on the entry timestamps information retrieved from the ingress router and the exit timestamps received from the egress router. It is expected that the IPNMS will poll each monitor once every sampling interval associated with the monitor.

The CORBA IDL specification of the *Edge_Metering_Monitoring* interface is given in [45].

Core Capacity Management Interface

This interface is provided by each core router. It is used by the IPNMS to query link capacity information from each core router as well as to change link capacities. When the IPNMS queries capacity information, the core router returns the capacity of each link (interface) connected to the router. For each such link, the capacity provisioned for each class is returned. To change the capacity of a link, the IPNMS identifies the link and supplies the new capacity information to the core router.

The CORBA IDL specification of this interface is given in [45].

4.6. *Interdomain QoS Management Software Architecture*

4.6.1. Proposed Solution Approach

We adopt a two-tier IPNMS structure for the NGI multidomain network management architecture to provide a modular design. There is one IPNMS in each autonomous system. An IPNMS is composed of *intradomain IPNMS module* (or simply intradomain IPNMS) and *interdomain IPNMS module* (or simply interdomain IPNMS). Requests for pipes are always addressed to the intradomain IPNMS. The latter can filter out requests for those pipes whose destinations are beyond the local domain, and forward these requests to the interdomain IPNMS for further processing. The intra-domain IPNMS functional units identified in Section 3.1, namely, the Setup/Release Unit, the Bandwidth Management Unit, the QoS Monitoring Unit and Accounting Information Management Unit, will be part of the proposed interdomain IPNMS design but with appropriate extensions and modifications to handle interdomain functionalities. Modifications and extensions are required because having a single management element at interdomain level is not only prohibitively expensive, but also contradictory to the concept of Autonomy. Therefore, the NGI Interdomain Network Management will adopt a distributed IPNMS structure with one IPNMS in each domain. Now, once the routes have been determined through both the underlying interdomain routing and intradomain routing protocols, and the request admission control procedure as well as the security and authentication functionalities have been completed, the Setup/Release Unit can function predominantly as before, but with the added details to meet the needs that arise at the border routers. The traffic policing and the accounting information management units will also be needed at the domain level to ensure service agreements between neighboring domains are honored.

While the proposed distributed IPNMS structure is generic enough to cope with any type of interdomain routing protocol, BGP-4 is assumed as the underlying interdomain routing protocol for its role of a de facto standard in today's Internet [46]. Under this assumption, in addition to those BGP-4 speakers established for fulfilling regular interdomain routing, it is also required that the IPNMS of each and every domain participate in BGP-4 execution. (Fortunately, the BGP-4 standard does not require the hosts executing the Border Gateway Protocol be routers. A non-routing host could use BGP to exchange routing information with a border router in another Autonomous System.) Every pair of IPNMSs of neighboring domains must maintain a reliable link (e.g. a TCP connection) between them. IPNMSs of neighboring domains will use this link to exchange not only network reachability information, but also service agreement updates and requests for setup/releasing IP pipe crossing multiple domains.

4.6.2. Service Agreement between Neighboring Domains

Complete discussion about policies governing traffic exchanged between neighboring domains is beyond the scope of this document. Therefore, we only concentrate on the items that are relevant to our concern. For easy discussion without losing generality, we assume two domains, D_A and D_B are connected by a link, L . We further assume that the links are full duplex. Hence, link L can be logically treated as two unidirectional links: one inbound and one outbound with respect to a domain. The interdomain IPNMS requires the bandwidth and its allocation between the two service classes be specified in the service agreement for each direction of the link using the following structure:

```
Struct Service_Agrmnt_Info {
    string Conn_ID_Interface;
    string Medium_Type;
    float Capacity;           // bits/second
    float Class1_bw;          // bits/second
    float Class2_bw;          // bits/second
};
```

The specifications of an interdomain link can be configured either centrally at the IPNMS or locally at the border router and then polled by the IPNMS, which is sorely an administrative decision of a domain. In any case, the IPNMS of D_A (D_B) is responsible to inform the IPNMS of D_B (D_A) the specifications for each inbound link between them. Note that the presence of Capacity in the Service_Agrmnt_Info

structure is necessary because a domain may not be willing/able to accept traffic at full link capacity from a neighbor.

4.6.3. Management Components and Interfaces

The intradomain IPNMS design is extended to the interdomain case by building upon the intra-domain components.

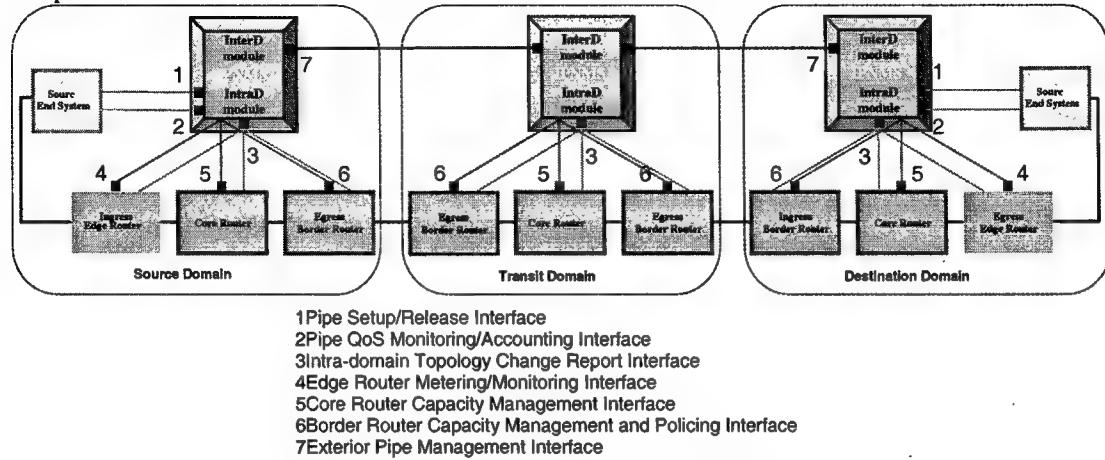


Figure 4-3 – End-to-end Setup of IP Pipes

Figure 4-3 depicts a typical system environment for two end systems setting up an IP pipe crossing multiple domains and where the IPNMSs fit in within such an environment. With help of the figure, terms with respect to an IP pipe crossing multiple domains are defined as following:

- **Source Domain:** The domain to which the source end system belongs.
- **Transit Domain:** A domain that provides routes for a portion of the pipe, but neither the source end system nor the destination end system belongs to this domain.
- **Destination Domain:** The domain to which the destination end system belongs.
- **Border Router:** A router of a domain that has a direct physical connection (which might be a shared medium) with another router of a different domain such that these two routers can exchange packets without resorting to interdomain or intradomain routing.

4.6.4. Management Components and Interfaces of the IPNMS

In addition to the subsystem interfaces found in intradomain IPNMS (i.e., Pipe Setup/Release interface, Pipe QoS Monitoring/Accounting interface, (intradomain) Topology Change Report interface, Edge Router Metering/Monitoring interface, and Core Router Capacity Management interface), two new interfaces are added. They are Border Router Capacity Management and Policing interface and Exterior Pipe Management interface. One may notice that we do not have any interface for reporting interdomain topology changes. This is because the IPNMS can learn it through BGP participation. Functions of the management components, namely Bandwidth Management Unit (BMU), pipe Setup/Release Unit (SRU), and QoS Monitoring/Accounting Information Management Unit (QAU), are enhanced to communicate with IPNMSs in neighboring domains to deal with the multiple domain nature of pipes spanning more than one domain.

Service and Bandwidth Management

To perform admission control and pipes setup and release, an IPNMS has to maintain required state information about connectivity to each of its neighboring domains as well as information about interior

topology of its own domain. While being a BGP-4 speaker facilitates the needs for routing information, a IPNMS also needs to know the aggregate bandwidth provisioning levels on a link to a neighboring domain as agreed upon with that domain, and the actual aggregate bandwidth occupancy levels for all supported traffic classes. Part of this second type of information is determined through the service agreement, and the rest relies on proper communication between the IPNMS and the border routers as well as the IPNMS of the neighboring domain. There is a BGP connection between IPNMSs of every pair of neighboring domains.

The Bandwidth Management Unit (BMU) of the IPNMS is responsible for managing bandwidth allocations on an end-to-end basis within an administrative domain. While setting up a pipe crossing multiple domains, it needs to perform some enhanced functions. It checks to see if a path can be established to accommodate the requested bandwidth and service class in accordance to the incoming request. It is able to do this since, first, it has full knowledge of the network topology interior to its domain and all links to its neighboring domains, and second, it knows the network status as well as the routing algorithm employed by the underlying network. Based on the specific routing algorithm, the BMU computes a route for the specified pipe request. While this route computation is end-to-end for a intradomain pipe, the BMU can only determine an egress border router for an interdomain pipe and has no controls beyond that. The checks for unallocated bandwidth are done with respect to only the portion of pipe interior to its domain.

Thus, we see that the Bandwidth Management Unit makes it possible to confine state information on an administrative domain basis within the IPNMS. This allows the IPNMS alone to make admission decisions, rather than having to involve the network routers no matter the requested pipe is interior to the domain or goes beyond it.

As mentioned earlier, the BMU stores an image of the network topology and link capacity information, among other things, in its own database. Since the available link capacity (which is based on the provisioning and current occupancy levels) will vary as a function of time, this configuration information will essentially be stored in the form of “soft” states within the BMU. However, since an IPNMS can only control its own side of each interdomain link, the BMU for interdomain links has to do two things. The first is managing outbound links, and the second is monitoring inbound links, both through communicating with the border routers in order to obtain the most updated status of the interdomain links, thereby refreshing the soft states contained in its database.

Internally, we propose the part associated with interdomain links in the database is structured as follows. The database maintains a list of the border routers of its domain. For each of the border routers, it maintains a list of the interdomain links together with the total link capacity, available link capacity, current occupancy and provisioning level information. Inbound links and outbound links are listed separately.

To begin with, the database entries for the link capacity information are as follows: the current occupancy levels are initialized to zero, the available link capacities and the provisioning levels are set to the ones as indicated by the service agreement. When a route computation for a multi-domain pipe is performed, the routing algorithm checks with the database for the current link occupancy and available capacity levels on the outbound link of the border router as well as on the entire interior route in order to decide whether to admit the pipe request. If the request is accepted, the available capacity values (current occupancy values) within the database are decremented (incremented) by an amount equal to the bandwidth being requested, on each of the links in the corresponding route. Of course, when the request is rejected, the database is not altered. Next, whenever the IPNMS receives any link failure notification for an outbound link, it sets the associated available link capacity value to zero and marks the status of corresponding link as down. In the case of a border router failure, it marks each of the associated links (be they inbound or outbound) as down and zeros the outbound links’ available capacities, and also marks the router status as down. In both cases, new routes are computed for those IP pipes that were affected by the failure event in question. The IPNMS will update its database accordingly. Finally, upon receipt of a recovery information, the IPNMS resets the link/node status and updates the available capacity of the associated links in its internal database.

In conclusion we observe that apart from providing bandwidth management functionalities, the BMU also provides information on the failure/recovery patterns within the network and on its boundaries since it logs

all the facility failures in its database in addition to maintaining the status of each of the network link and router. This information can be used by network providers/engineers to appropriately dimension their network and interconnection with neighboring domains, and to determine what kind of survivability mechanisms need to be installed/updated.

Setup and Release Multi-domain IP Pipes

The philosophy of managing interdomain pipes is the same as that in intradomain case. Maintaining full knowledge about status of all interdomain links, the IPNMS can perform admission controls on any incoming request for an IP pipe seeking resources from its domain with minimum involvement of routers along the pipe. Similar to edge routers in the intradomain case, border routers are required to perform additional functions in the operation of pipe setup and release. These functions are essentially related to monitoring and policing traffic on an aggregated basis in accordance to the service agreement with the traffic-injecting domain.

In setting up an interdomain IP pipe, the IPNMSs in the involved domains perform the following operations.

In the Source Domain

An end system requests the IPNMS to set up an IP pipe by invoking the `Setup_Pipe` operation. The end system identifies the end points of the pipe using a `Classifier` structure that contains the host IP address and port number of the source and the destination end points. For a multi-domain pipe, the destination is in another domain generally not neighboring with the source domain. Hence, the IPNMS first needs to locate the border router to the next-hop domain leading to the destination through the best route subject to whatever policies being enforced based on its interdomain routing knowledge learned through participating in the interdomain routing protocol (e.g. BGP-4). Then the Bandwidth Management Unit in the IPNMS is consulted for current provisioning levels of resources along the route. If the BMU finds the request can be accommodated, the `Setup/Release` Unit of the IPNMS records this pipe in its internal database by creating a table entry that is designed to hold the following information:

- `PipeClassifier (<ingress_adr, next-hop_adr, pipe_spec>)`
- `PipeSpec (<class, bandwidth, burst size, duration>)`
- `PipeID`, and
- `MonitorSpecs ([<delay-monitor-id, delay-value>, <bandwidth-monitor-id, bandwidth-usage-value>, <loss-monitor-id, loss-value>])`

In the case of the first database information field, i.e. `PipeClassifier`, the *egress-border (ingress-border)* entry will be marked empty in the case of a source (destination) domain. The value of the third database information field, i.e. `PipeID`, is assigned¹¹ by the IPNMS of the source domain, and is to be used by all IPNMSs associated with the pipe so that any pipe will be identified by a unique ID throughout the IPNMSs that are supporting the pipe.

Meter setup is the same as in the intradomain case.

So far, the portion of the pipe interior to the source domain, i.e. from the source end system to the selected egress border router, has been setup, in pretty much the same way as in the intradomain case as if the selected border router were the destination end system. While the egress border router is selected by the BGP, route computation for this portion of the pipe is based on the underlying intradomain routing protocol. To continue setting up the rest portion of the pipe, the IPNMS of the source domain now has to

¹¹ In order to make sure a `PipeID` unique throughout all the pipe crossed domains, it is suggested that the `PipeID` be created as a function of the pipe's end systems' IDs and port numbers.

inform the IPNMS of the next-hop domain about the requested pipe through the Exterior Pipe Manage Interface. This interface functionally is the same as the Pipe Setup/Release Interface of the IPNMS to an end system, except that the `PipeClassifier` structure is the augmented to include the interdomain characteristics of the pipe as well as its specifications.

In comparison with the `Classifier` definition in the intra-domain case, we see that border routers are added in `PipeClassifier`. While the value of `ingress_adr` is learned from the IPNMS of the previous domain, the value of `next-hop_adr` is obtained from Loc-RIB, one of the Routing Information Bases maintained by all BGP-4 speakers [41]. This expanded definition enables the IPNMS of the next-hop domain to treat the ingress border router as the ingress edge router as far as setting up this pipe is concerned, which maximizes the reuse of intradomain functions for multiple domain pipes management.

The following list summaries the actions taken by the source IPNMS in response to the multidomain IP pipe setup request.

- Determine the egress border router.
- Create the corresponding `PipeClassifier`.
- Check resource availability from the ingress edge router to the selected egress border router. Check resource availability on the outbound link on the egress border router.
- If the resource is not sufficient to accommodate the new pipe, return a reject to the end system.
- Otherwise, reserve the required resource and inform the IPNMS of the next-hop domain through the Exterior Pipe Manage to setup the rest part of the pipe.
- Wait for reply from the IPNMS of the next-hop domain.
- Forward the reply to the end system.
- If the reply is a reject, cleanup database entries corresponding to the pipe.
- If the reply is admittance, invoke `Setup_Meter_Entry` at ingress edge router.
- Update pipe database.

In a Transit Domain

Upon receiving the request on its Exterior Pipe Manage Interface, the IPNMS of the next-hop domain of the source domain (in general a transit domain) will work out the portion of the pipe interior to its domain. For interior route computation purpose, the `PipeClassifier.NEXT_HOP_adr` entry is interpreted as the ingress edge router, and the egress border router determined by the IPNMS participated interdomain routing protocol is counted for the egress edge router to the destination end system. Upon positive response from the BMU for resources checks, the SRU updates its resource database, records this pipe in its internal database, forwards the pipe request to the IPNMS of the next-hop domain, and notifies the IPNMS of the previous-hop domain that the pipe request is honored.

The following list summaries the actions taken by the IPNMS of a transit domain in response to the multidomain IP pipe setup request received from the previous-hop domain's IPNMS.

- Determine the egress border router.
- Modify the corresponding `PipeClassifier`¹².

¹² The modification is done like this:

```
PipeClassifier.ingress_adr = PipeClassifier.next-hop_adr;  
PipeClassifier.next-hop_adr =  
    <address at the other end of the outbound link at the selected egress border router>;
```

- Check resource availability from the `PipeClassifier.ingress_adr` to the selected egress border router. Check resource availability on the outbound link on the egress border router.
- If the resource is not sufficient to accommodate the new pipe, return a reject to the IPNMS of the previous-hop domain.
- Otherwise, reserve the required resource and inform the IPNMS of the next-hop domain through the Exterior Pipe Manage to setup the rest part of the pipe.
- Wait for reply from the IPNMS of the next-hop domain.
- Forward the reply to the IPNMS of the previous-hop domain.
- If the reply is a reject, cleanup database entries corresponding to the pipe.
- Update pipe database.

In the Destination Domain

Things are most straightforward here. What the IPNMS concerns is the portion of pipe between the ingress border router to the destination end system and monitors setup at the egress edge router. While the positive acknowledgement has to be made by the IPNMS of the destination domain, a negative one (i.e. reject a pipe request) is going to be made by the IPNMS of the first domain that does not have sufficient resources to accommodate the requested pipe. Such a resource shortage domain can be any one of the source domain, a transit domain, or the destination domain. When a new pipe is successfully admitted, the destination IPNMS

The following list summaries the actions taken by the destination IPNMS in response to the multidomain IP pipe setup request received from the previous-hop domain's IPNMS.

- Modify the corresponding `PipeClassifier`¹³.
- Check resource availability from the `PipeClassifier.ingress_adr` to the ingress edge router.
- Return an admittance (a reject) if the resource is (in)sufficient to accommodate the new pipe to the IPNMS of the previous-hop domain.
- Update pipe database.

The pipe release procedure is relatively simple. Each related SRU simply removes from its internal database the table entry created for the pipe being released, updates its resource database, and free up the memory. Note that the event of releasing a pipe may be initiated by an end system, or caused by other difficulties incurred by any network element associated with the pipe. Therefore, when a related IPNMS decides to remove the table entry of a pipe, it must transmit this decision to the IPNMSs of both upstream and downstream domains with respect to the pipe to let them also release their resources for this broken pipe.

The pipe release procedure is also used to tear down a partially setup pipe. To be more specific, assume one transit domain could not support a requested multi-domain pipe due to insufficient resource, all the IPNMSs of the previous-hop domains (up to the source domain) should be notified of this so that they can free up the already reserved resources. The rejection to a pipe request can be handled in same way by the IPNMS as when the pipe is being released.

Management Components and Interfaces in Border Routers

¹³ The modification is done like this:

`PipeClassifier.ingress_adr = PipeClassifier.next-hop_adr;`

In the proposed interdomain QoS management architecture, the border routers and the interior routers of a domain play different roles. This is because, for the scalability of the architecture, interior routers are discharged from the complex QoS functions and only left with efficient forwarding coupled with minimal service differentiation activity.

Since border routers take a different role than that of either the core routers or edge routers, a new type of interface with IPNMS, called Border Router Capacity Management and Policing Interface, is defined. The functions provided by border routers on inbound links include packet classification based on traffic classes, policing incoming traffic based on the service agreement with the traffic injecting domain, and monitoring the inbound traffic and collecting information for accounting purpose. It is optional to provide shaping functions applied to aggregated traffic on outbound links. While all of these functions are performed on aggregated traffic, border routers do have capability to identify packet associated with a specific pipe. Such a capability is necessitated by the need of QoS debug when certain pipes experience QoS degradations caused by problems undiagnosable at aggregated traffic level.

Capacity Management

It has been justified the need for an interface between the intra-domain IPNMS and a core router to gather the connectivity and capacity information. The same argument is true for the border routers. For each border router, the IPNMS is able to do the following operation:

```
Poll_Capacity(Service_Class)
  -> (Conn_ID_Interfaces, Capacity, Medium_Types)
```

where Conn_ID_Interfaces are the list of directly connected interdomain interfaces. Medium_Types are necessary to figure out how much of the Capacity is in fact available to the connected interdomain interfaces for the specified Service_Class. For instance, if it is a shared medium, then the capacity is not exclusively available, rather is shared with others.

In the reverse direction, a border router might want to report its connectivity¹⁴ and capacity information to IPNMS. System initialization, and interface up/down events (due to failure and recovery, respectively) are examples of such occasions. The corresponding operation is:

```
Report_Capacity(Service_Class, Conn_ID_Interfaces, Medium_Types)
  -> (Capacity)
```

When IPNMS decides to change the capacity of an interface, or enable/disable an interface (due to changes of service agreement with related domains), it can use the following interface:

```
Change_Capacity(Service_Class, Conn_ID_Interfaces, Capacity)
  -> (Status)
```

Currently, the most popular and most primitive interdomain routing is Border Gateway Protocol (BGP). When routes connecting domains change, the IPNMS being a BGP-4 speaker can naturally learn of the changes, and it can update its interdomain routing table accordingly. But note that the BGP only provides the IPNMS of the network layer reachability information (NLRI). The capacity information regarding the interdomain links provided by border routers should complement the route information for the IPNMS to manage the IP pipes.

Finally, the CORBA IDL specifications of the Border Router Capacity Management and Policing interface and of the Exterior Pipe Manage interface are in [45].

¹⁴ Connectivity information may also be exchanged through the underlying interdomain routing protocol, e.g. in the BGP-4 case, the IPNMS is also a BGP-4 speaker.

4.6.5. Merits of the proposed approach

Though BGP-4 is mentioned frequently in discussion, the proposed interdomain QoS management architecture is a generic solution in a sense that it does not rely on any specific interdomain routing protocol. This and other merits are listed below:

- Generic solution, independent of routing protocol.
- Efficient pipe operation with limited involvement of selected network entities (only end systems, edge routers and IPNMSs are involved under normal operations; border routers are involved only in QoS troubleshooting).
- Scales well to large multi-domain environment because only IPNMSs of neighboring domains interact with each other.
- Domain-crossing is transparent to end systems.

4.6.6. Limitations of the Proposed Approach

This approach has some limitations: The proposed interdomain IPNMS architecture requires all domains between and at both ends of an IP pipe be participating in the algorithm. Tunneling schemes to cross “interdomain IPNMS unaware” domains are not considered. So far, only point-to-point pipes are supported. It would be nice if multicast functions are supported in future work.

4.7. A Comparison of QoS Mechanisms Used in ATM and Proposed NGI IPNMS

Having completed a description of the proposed network management architecture that supports QoS for IP-based applications, it is interesting to compare this approach with the ATM approach to QoS. Table 4-3 presents such a comparison.

Comparison Aspect	ATM	IPNMS
Admission control and resource allocation	Resource allocation and admission control decisions are performed in every switch.	Admission control decision is made only by the IPNMS. Decision based on router connectivity and link load information maintained by the IPNMS. Packet scheduling in routers is based on class priority (Class 1 and Class 2). Edge routers classify packets and mark the priority field.
Route selection	Each switch selects the route for a connection based on bandwidth availability on links and QoS attributes such as delay and loss. Source routing is also possible.	Relies on the routing protocol used in the routers. Currently used Intradomain routing protocol, i.e., OSPF, does not use QoS parameters in route selection. Source routing in IP involves too much overhead, since each packet then needs to carry the source route information.
Policing	Policing only at the edge switches and border switches.	Policing only at the edge routers and border routers.
Connection/Pipe information	State maintained in every switch.	Pipe information is maintained only in the IPNMS and the edge routers. Only the

		IPNMS and the edge routers are involved in pipe setup and release operations.
Connection/Pipe restoration upon network failures	Needs explicit restoration to recover from failures. Restoration initiated by switches. The initiating switch may be the switch that detects the failure or the switch that originated the connection setup.	When a router/link fails, the topology change is communicated to all routers, new routes are computed, and packets are rerouted along new routes. IPNMS determines the set of pipes affected by the failure, and verifies if the required QoS can be provided to the pipes along the new route. It determines the set of pipes whose QoS has now degraded, and either releases them or sends a QoS degraded notification to the end systems depending on the policy agreed at pipe setup time.
Failure of the IPNMS		Two types of restoration schemes are possible. In one scheme, the IPNMS maintains the pipe information in a crash-resistant (stable disk) storage and uses it upon recovery. In another scheme, the IPNMS polls all edge routers and collects source-destination information about all pipes that it established prior to its failure. It then computes the route for each pipe and reconstructs the link bandwidth information.

Table 4-3 – Comparison of QoS Mechanisms Used in ATM with Proposed NGI IPNMS

5. Conclusions

We conclude this report with a summary of our results on developing a network architecture and network management approach for the NGI backbone that support the levels of QoS and survivability required by envisioned NGI applications.

5.1. Network Architecture Results

To guide this work, we first identified two key application types that are expected to impose the most exacting demands on the NGI backbone network. These are Distributed Interactive Simulations, which involve the creation of a synthetic battlefield environment, and Global Command and Control Systems, which must be able to move a fighting force around the globe and provide the information it needs to complete its mission. Keeping the general characteristics of these two types of applications in mind, we then elaborated a list of network performance-related requirements for the NGI backbone network. Coupling these performance needs with the requirements for survivability, QoS, scalability, and interoperability that were delivered in our previous reports, we developed functional specifications for an NGI backbone architecture with both unicast and multicast capabilities. The primary focus of this paper has been on elaborating this functional architecture.

Three key functional groups comprise the top level in the architectural hierarchy we have described here, and each of these was presented in detail. To summarize, the Information Transport group consists of functions that relate to transporting information across the network. Accordingly, its key function components are transmission, interworking, and switching. The Network Control group consists of functions related to the real-time control of network operations; these include signaling, QoS control, routing control, and reconfiguration control. Finally, the Network Management functional group consists of

component functions for managing network configuration, network performance, network faults, and network security.

As discussed, network management functions are required to interact closely with network control functions in this architecture to ensure that information can be transported across the network with the QoS required by the associated end applications. In a separate task force, Bellcore has been conducting studies on various aspects of network management for the NGI backbone network. In the present report, we therefore focused on the functional demands that relate to information transport and network control. We have also provided a discussion of the reconfiguration strategies applicable to an IP/WDM NGI backbone Network in an appendix.

Finally, we delineated the key functional components required for AutoPops in the NGI functional architecture— that is, of the interconnection points between the backbone subnetworks and between those networks and the attached NGI site networks. The functional components of an AutoPop include auto-configuration, fault detection, performance monitoring, QoS control, signaling, routing control, mobility management, reconfiguration control, switching, interworking, and transmission. Each function was described in detail.

We have developed not only the functional specifications for the NGI backbone based on a unicast reference model, but also for supporting IP-over-WDM multicasting in the NGI network. The rationale behind multicast is to maximize bandwidth usage (and minimize waste) by aggregating traffic on the common paths. To accomplish the goal of multicasting, group management and routing management are among the most important functional requirements. We identified and defined new multicast-specific functional components that are needed for multicast communication in the NGI system, and also pointed out any modifications that are required from our definitions of functional components for the unicast model.

Although WDM offers a highly reliable, ultra high-speed communication channel, the current state of WDM technology makes an implementation of WDM-layer multicast routing infeasible at present. Slow reconfiguration times and a limit on the number of wavelengths supported by each fiber are some of the major technological hurdles that must be crossed before scalable WDM-layer multicasting can be realized in the NGI network environment. However, in this deliverable we proposed and evaluated four alternative ways in which WDM multicast routing can be implemented in the NGI environment.

IP multicast can be used in the NGI network without functional modifications in WDM subnetworks, but it suffers from bandwidth waste. *Broadcast & Select (B&S)* is a viable option under large-scale multicast groups with geographical spreads, but can waste communication bandwidth under sparse multicast groups. For applications with strategic and semi-dynamic group management, *Multiple Path Configurations (MPC)* is most promising given the current state of WDM technology, with one of its greatest strength being the potential to minimize bandwidth waste. In MPC, the AutoPops retain pre-computed path configurations that are subsets of the multicast spanning tree. The proper path configuration is chosen from this set when there is a change in group membership, and the configuration is switched accordingly.

When properly designed and implemented, the MPC method can provide a bandwidth-efficient multicasting environment for the class of applications in which the receiving group is semi-static and group updates occur with regular geographic patterns. However, proper design entails: (1) finding efficient ways to estimate the multicast spanning tree, and (2) designing appropriate algorithms for building path configurations that correspond to subsets of the spanning tree, as well as for selecting the right path configurations when a new member (or a group of members) joins or leaves a given group.

Finally, we described the method of *WDM Multicast Routing (WMR)*. The other three routing methods are designed to cope with current WDM limits. But WMR is aimed at next-generation WDM technologies that will be able to provide fully functional networking capabilities, including fast reconfigurations, WDM-layer headers or tags, and switch buffers (possibly in optical domain).

Implementing NGI multicast capabilities has implications for unicast communications in terms of both the functional architecture and network performance. A new set of functions such as session, group, and routing management needs to be defined to support NGI multicast based on IP-over-WDM technology. In addition, some functional components (such as routing and reconfiguration control, network control signaling, QoS and security control) need to be modified or enhanced from their definitions in the unicast

reference model. Nonetheless, unicast applications will also benefit when fast-reconfigurable switches are available for implementing WDM-layer multicasting. Such benefits will include the ability to provide dynamic switching, better data synchronization for isochronous multimedia streams, and faster adjustment of the network state depending on current demand.

In summary, we have presented a comprehensive set of functional descriptions for the NGI backbone network architecture. Throughout our detailed discussions in this report, we have described required functionalities independent of any implementation limitations. The intention is that these functional specifications can serve productively as planning guidelines for future implementations of the NGI backbone network.

5.2. Network Management Results

The key features of the NGI network management architecture described in this document can be summarized as follows:

1. An application layer QoS model has been proposed. This model categorizes applications into two classes: continuous communication class and discrete communication class. The continuous communication class is used to model producer-consumer interactions between applications. Such interactions typically occur in the exchange of audio or video information between applications. The discrete communication class models client-server (request-reply) interactions between applications. For each category of application, QoS parameters that can be used to specify the QoS requirements of the application are identified.
2. An IP layer QoS model based on the differentiated services approach has been proposed. The IP layer traffic is classified into two classes: Class 1 traffic is delay sensitive requiring deterministic delay guarantees (typically CBR traffic); Class 2 traffic is delay tolerant but relatively sensitive to loss of information (typically VBR traffic). The delay guarantee provided by the network to a Class 1 traffic is with reference a traffic rate agreed between the network and the traffic source. Class 1 traffic from a source that exceeds the agreed traffic rate is rejected by the network. Class 2 traffic is provided a minimum bandwidth guarantee, but traffic above the minimum rate is accepted by the network and transported on a best-effort basis. The merits of the proposed IP layer QoS model are that it caters to applications with varying degrees of QoS requirements and it scales to large sized IP networks.
3. A software architecture for intra-domain QoS management has been developed. The software architecture consists of four subsystems, i.e., End System, IPNMS, Edge Router, and Core Router. The software architecture was specified using CORBA IDL [45] and the IDL specifications can be mapped to other management protocols as needed. The proposed QoS management software architecture has the following distinguishing features:
 - End-to-end QoS for streams is provided through the cooperation of end systems and the management components in the network.
 - It enables end systems to make efficient use of QoS capabilities provided by the network. The network supports IP pipes between end systems. QoS for IP pipes is provided through a combination of link capacity provisioning and class based packet scheduling in the core network, bandwidth management for IP pipes by the IP Network Management System, and policing and class based packet scheduling in the edge routers.
 - It does not require routers to participate in any reservation protocol for setting up pipes, this architecture scales well for large networks that may be required to support thousands of pipes concurrently.
 - The decision to admit a pipe rests only with the IPNMS. Only an end system, the IPNMS, and the edge routers need to interact in setting up a pipe. Hence, the pipe setup operation can be implemented in a very efficient manner.
 - The core routers have no knowledge of pipes. Core routers are concerned only with fast forwarding of packets taking into account class priorities. QoS support in the backbone network is through a combination of network provisioning and packet scheduling.

- The architecture makes a clear separation between three levels of QoS mechanisms: fine grain stream level QoS support mechanisms, medium grain pipe level QoS support mechanisms, and coarse grain traffic class level QoS support mechanisms. Middleware in the end systems cooperate and collect stream specific QoS information. The IPNMS interacts with edge routers and collects pipe specific QoS information and delivers such information to the end systems on demand. The pipe specific QoS information collected by the IPNMS can be used by the middleware to adapt its decisions concerning mapping of streams to pipes. The IPNMS also collects traffic class specific QoS information from the edge and core routers. Such information can be used by network engineers and planners for network reconfiguration and evolution.

4. An inter-domain management architecture for IP networks has been developed. The architecture consists of two parts, i.e., QoS management and subscription and security management. The proposed inter-domain management architecture has the following distinguishing features:

- End-to-end QoS for IP pipes is provided through the cooperation of IPNMSs of the neighboring domains.
- It enables end systems to make efficient use of QoS capabilities provided by the network. The architecture supports IP pipes spanning multiple domains. QoS for IP pipes is provided in two perspectives. For the portion of a pipe interior to a domain QoS is supported through a combination of link capacity provisioning and class-based packet scheduling in the core network, bandwidth management for IP pipes by the IP Network Management System (IPNMS), and policing and class-based packet scheduling in the edge routers. For the portion of the same supported by inter-domain links, QoS is supported through domain-level traffic monitoring and policing in accordance with the service agreement between the domains on both ends of the inter-domain link in question.
- It does not require border routers to participate in any reservation protocol for setting up pipes. This architecture scales well for large networks that may be required to support thousands of pipes concurrently.
- All services require subscription in order to protect the services against common security threats. CORBA security services are adopted to tackle the security threats threatening interactions between users and the IPNMS, and SNMPv2 security services are deployed to fight security threats that interactions between IPNMS and the underlying network may face.

5. A scheme for configuring an IP network using a WDM network for the link layer has been proposed. The proposed scheme is a variant of the NHRP scheme proposed in IETF. Taking into account the limits of current-day WDM technology, it is proposed that an IP network be configured as several subnetworks, where each subnetwork has several hosts and routers. One of the routers in each subnetwork is designated as the access router. Hosts and the access router establish WDM connections between them and use these connections for transporting packets. Normally, packet transport between hosts in a subnetwork occur via the access router, and packet transport between hosts in different subnetworks occur via several routers. But, hosts can establish WDM connections between them, if needed, and use these connections to transport packets between them directly by-passing routers. Establishment of such host-host connections requires an address resolution service that maps the IP address of a host to a set of WDM port addresses of the host. It has been proposed that the access routers perform this service for the hosts in the local subnetwork in cooperation with access routers of other subnetworks.

6. Finally, a software architecture for inter-domain WDM network management has been developed. This architecture uses the notion of federation to group the domains that want to share resources among themselves for certain business arrangement. Resources management at federation level, with emphasis on management of federation topology, multi-domain connections and inter-domain links, is the innovative part of the work.

The NGI network management architecture described in this document is an initial design. It has addressed only a subset of integrated management capabilities. Further work is needed to extend this architecture and

design to provide additional management capabilities. Currently, the following management capabilities have been identified as problems that need to be addressed in future work:

- **Integrated IP-WDM QoS management:** The major challenge here is how to match IP layer QoS parameters with WDM layer QoS parameters which are related with optical characteristics. The impact of different WDM network architectures, ranging from provisioned WDM connections, signaling based WDM connection management, to burst switching architectures, on IP layer QoS management needs to be investigated.
- **Dynamic QoS:** The QoS models that have been proposed in this document for the IP layer and the application layer do not support changes in the QoS parameters of an existing application stream or IP layer traffic. This limitation should be overcome in the future work.
- **QoS Sustenance:** The IPNMS design described in the report does not take any explicit action (in terms of informing the end systems) when a network facility failure occurs and when the pipe affected by the failure event cannot be re-routed/re-established. Instead, this is reflected as a severe degradation in the QoS of the associated pipe, which the end systems eventually discover by querying the IPNMS (either periodically or sporadically) for QoS information. One possible future work item is to let the IPNMS record any degraded QoS in the re-routed/re-established path for the affected pipes and communicate this information to the end system as soon as such an event occurs via a new interface, rather than have the end-system eventually learn of the degraded QoS via its querying operation.
- **Multipoint communication:** The QoS model and mechanisms developed in this work support only point-to-point communication. An important further work is to extend the proposed design for multipoint communication.

The implementation of this NGI network management architecture as well as some of the additional work described above are currently being undertaken in a separate research program [47].

6. List of Acronyms

ATM	Asynchronous Transfer Mode
AutoPop	Autoconfiguration Point of Presence
BGP	Border Gateway Protocol
BMU	Bandwidth Management Unit
CBT	Core-Based Trees
CORBA	Common Object Request Broker Architecture
CR	Core Router
DIS	Distributed Interactive Simulation
DVMRP	Distance-Vector Multicast Routing Protocol
ER	Edge Router
ES	End System
GCCS	Global Command and Control System
IDL	Interface Definition Language
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPNMS	IP Network Management System
NE	Network Element
NGI	Next Generation Internet
OSPF	Open Shortest Path First
OSI	Open Systems Interconnect
PIM	Protocol Independent Multicast
QAU	QoS Monitoring/Accounting Information Management Unit
QoS	Quality of Service
RSVP	Resource Reservation Protocol
SONET	Synchronous Optical Network
SNMP	Simple Network Management Protocol
SRU	Setup/Release Unit
ST2	Stream Protocol Version 2
VCI	Virtual Circuit Identifier
VPI	Virtual Path Identifier
WDM	Wavelength Division Multiplexing

7. References

- [1] Next-Generation Internet Initiative Concept Paper, July, 1997 (available from <http://www.ngi.gov>)
- [2] D-P. Hsing, R. Talpade, L. Kant, and T-H. Wu, "Assessment of Alternative NGI Backbone Network Architectures," Deliverable 4.2.1, January, 1998.
- [3] L. Kant, H. Kim and D-P. Hsing, "Modeling and Simulation Study of the Survivability of Candidate Architectures for the NGI Backbone Network Architectures", Deliverable 4.2.2, April, 1998.
- [4] See <http://www.ngi.gov/apps> for examples
- [5] DARPA Advanced Distributed Simulation (ADS) Program, <http://www.iso.darpa.mil>
- [6] GCCS/DII COE System Integration Support Technical Report/Study: GCCS Strategic Technical Architecture, February 27, 1997, <http://204.34.175.79/onlinedocs.html>
- [7] P. T. Brady, "Effects of Transmission Delay on Conventional Behavior on Echo-Free Telephone Circuits," *Bell System Technical Journal*, Volume 50, 1971.
- [8] D. E. Comer, *Internetworking with TCP/IP: Principles, Protocols, and Architecture*, Vol. 1, 1991, Prentice Hall.
- [9] D. E., Comer, *Internetworking with TCP/IP: Design, Implementation, and Internals*, Vol. 2, 1991, Prentice Hall.
- [10] W. R. Stevens, *TCP/IP Illustrated*, Vol. 1, 2, 3, Addison Wesley Publishing Company. 1994–1996.
- [11] Sun Microsystems Implementation Data
- [12] A. Ganz, "Implementation Schemes of Multihop Lightwave Networks", *Journal of Lightwave Technology*, Vol. 11, No. 5/6, May/Jun 1993.
- [13] J. Powers, *An Introduction to Fiber Optic Systems*, Irwin, 1997.
- [14] J. Bannister, J. Smart and A. Willner, " WDM IP Flow Technology (SWIFT)," *Proceeding of IEEE Gigabit Networking Workshop GBN'98*, March 1998.
- [15] T-H. Wu, *Fiber Network Service Survivability*, Artech House, 1992.
- [16] G. Meempat and L. Kant, "QoS Engineering Architecture for the Next-Generation-Internet," Deliverable 4.1.1, November, 1997.
- [17] D. Clark and J. Wroclawski, "An Approach to Service Allocation in the Internet", IETF Internet Draft, July, 1997.
- [18] K. Nichols, V. Jacobson and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", Internet Draft, draft-nichols-diff-svc-arch-00.txt, November 1997.
- [19] J. Wroclawski, "The Use of RSVP with IETF Integrated Services", IETF Internet Draft, July, 1997.
- [20] L. Zhang and S. Berson and S. Herzog and S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," RFC 2205, September 1997. (Status: Proposed Standard)
- [21] J. Heinanen, "Use of the IPv4 TOS Octet to Support Differential Services", IETF Internet Draft, October, 1997.
- [22] C. Huitema, *Routing in the Internet*, Prentice Hall, 1995.
- [23] M. Steenstrup, *Routing in Communications Networks*, Prentice Hall, 1995.
- [24] Tsong-Ho Wu, Gitae Kim, Deh-Phone Hsing, Latha Kant, Will Leland and Deanna Wilkes-Gibbs, "NGI Backbone Functional Specifications", Deliverable 4.2.3, September, 1998.

[25] R. Braudes and S. Zabele, "Requirements for Multicast Protocols," RFC 1458, May 1993. (Status: Informational)

[26] D. Kosiur, *IP Multicasting*, John & Wiley & Sons, Inc., Danvers, MA, 1998.

[27] J. Nishikido, et al., "Multiwavelength Securely-Authenticated Broadcast Network," *ECOC '97*, Vol. 4, pp. 17-20, Sept., 1997.

[28] W. Fenner, "Internet Group Management Protocol, Version 2," Internet Drafts, *draft-finlayson-mafp-00.txt*, January 17, 1997.

[29] S. Deering, et al., "An Architecture for Wide-Area Multicast Routing," *Proceedings of ACM SIGCOMM '94*, pages 126-135, London, England, August 1994.

[30] T. Ballardie, P. Francis and J. Crowcroft, "Core Based Trees (CBT) – An Architecture for Scalable Inter-Domain Multicast Routing," *Proceedings of ACM SIGCOMM '93*, pages 85-95, Ithaca, N.Y., USA, September 1993.

[31] D. Waitzman, et al., "Distance Vector Multicast Routing Protocol," RFC1075, November 1988.

[32] L. Kant, N. Natarajan, "PSOS Intra-Domain Network Management", Deliverable 4.3.1, December, 1997.

[33] N. Natarajan, C. Liu, T. Li and A. Roy, "PSOS Inter-Domain Management Architecture", Deliverable 4.5.1, July, 1998.

[34] P.G.S.Florissi and Y. Yemini, *Management of Application Quality of Service, Proceedings of the Fifth IFIP/IEEE International Workshop on Distributed Systems, Operations and Management*, October 1994.

[35] N. Natarajan, C. Liu, T. Li, A. Roy, *PSoS Inter-Domain Management Architecture*, Deliverable 4.5.1, July 1998.

[36] Jean-Francois Huard and A. A. Lazar, *On End-to-End QoS Mapping*, IFIP 1997, Chapman & Hall Publishers.

[37] Jean-Francois Huard and A.A. Lazar, *On QoS Mapping in Multimedia Networks*, Technical Report, Center for Telecommunications Research, Columbia University, New York.
<http://comet.ctr.columbia.edu/~{jfhuard.aurel}>.

[38] J.A. Zinky, D.E.Bakken and R.D.Schantz, *Architectural Support for Quality of Service for CORBA Objects, Theory and Practice of Object Systems*, Vol. 3(1), 1997.

[39] A. Campbell, G. Coulson, and D. Hutchison, *A Quality of Service Architecture*, Dept.of Computing, Lancaster University, U.K.

[40] R. Braden et.al, *Resource ReSerVation Protocol (RSVP): Version 1: Functional Specification*, IETF Internet Draft, July 1997.

[41] L. Delgrossi and L.Berger, *Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+*, RFC 1819, August 1995.

[42] C. Topolcic, Editor, *Experimental Internet Stream Protocol, Version 2 (ST-II)*, RFC 1190, October 1990.

[43] P.G.S.Florissi and Y. Yemini, *Management of Application Quality of Service, Proceedings of the Fifth IFIP/IEEE International Workshop on Distributed Systems, Operations and Management*, October 1994.

[44] *Control and Management of Audio/Video Streams*, OMG RFP Submission, Telecom/97-05-97, Version 1.2, October 1997.

- [45] N. Natarajan, R. Chadha, L. Kant, H. Kim, C. Liu, T. Li and A. Roy, "PSOS Network Management Architecture, Deliverable 4.5.2, August, 1998.
- [46] Y. Rekhter, T. Li, *A Border Gateway Protocol 4 (BGP-4)*, Internet Working Group, RFC1771.
- [47] Supernet Network Control and Management, contract F30602-98-C-0202.

DISTRIBUTION LIST

addresses	number of copies
AFRL/IFGA ATTN: ROBERT KAMINSKI 525 BROOKS ROAD ROME, NEW YORK 13441-4505	5
BELLCORE 445 SOUTH STREET MCC 15206R MORRISTOWN NJ 07960	5
AFRL/IFOIL TECHNICAL LIBRARY 26 ELECTRONIC PKY ROME NY 13441-4514	1
ATTENTION: DTIC-DCC DEFENSE TECHNICAL INFO CENTER 8725 JOHN J. KINGMAN ROAD, STE 0944 FT. BELVOIR, VA 22060-5218	2
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
ATTN: NAN PFRIMMER IIT RESEARCH INSTITUTE 201 MILL ST. ROME, NY 13440	1
AFIT ACADEMIC LIBRARY AFIT/LDR, 2950 P STREET AREA B, BLDG 642 WRIGHT-PATTERSON AFB OH 45433-7765	1
AFRL/MLME 2977 P STREET, STE 6 WRIGHT-PATTERSON AFB OH 45433-7739	1

AFRL/HESC-TDC
2698 G STREET, BLDG 190
WRIGHT-PATTERSON AFB OH 45433-7604

1

ATTN: SMDC IM PL
US ARMY SPACE & MISSILE DEF CMO
P.O. BOX 1500
HUNTSVILLE AL 35807-3801

1

CDR, US ARMY AVIATION & MISSILE CMO
REDSTONE SCIENTIFIC INFORMATION CTR
ATTN: AMSAM-RO-DB-R, (DOCUMENTS)
REDSTONE ARSENAL AL 35898-5000

2

REPORT LIBRARY
MS P364
LOS ALAMOS NATIONAL LABORATORY
LOS ALAMOS NM 87545

1

ATTN: D'BORAH HART
AVIATION BRANCH SVC 122.10
FOB10A, RM 931
800 INDEPENDENCE AVE, SW
WASHINGTON DC 20591

1

AFIWC/MSY
102 HALL BLVD, STE 315
SAN ANTONIO TX 78243-7016

1

ATTN: KAROLA M. YOURISON
SOFTWARE ENGINEERING INSTITUTE
4500 FIFTH AVENUE
PITTSBURGH PA 15213

1

USAF/AIR FORCE RESEARCH LABORATORY
AFRL/VSOSA(LIBRARY-BLDG 1103)
5 WRIGHT DRIVE
HANSOM AFB MA 01731-3004

1

ATTN: EILEEN LADUKE/D460
MITRE CORPORATION
202 BURLINGTON RD
BEDFORD MA 01730

1

OUSD(CP)/DTSA/DUTD
ATTN: PATRICK G. SULLIVAN, JR.
400 ARMY NAVY DRIVE
SUITE 300
ARLINGTON VA 22202

1

***MISSION
OF
AFRL/INFORMATION DIRECTORATE (IF)***

The advancement and application of information systems science and technology for aerospace command and control and its transition to air, space, and ground systems to meet customer needs in the areas of Global Awareness, Dynamic Planning and Execution, and Global Information Exchange is the focus of this AFRL organization. The directorate's areas of investigation include a broad spectrum of information and fusion, communication, collaborative environment and modeling and simulation, defensive information warfare, and intelligent information systems technologies.